

MULTI-INPUT MULTI-OUTPUT (MIMO) DETECTION BY A COLONY OF ANTS

A Thesis

by

DANA N. JABER

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2006

Major Subject: Electrical Engineering

MULTI-INPUT MULTI-OUTPUT (MIMO) DETECTION BY A COLONY OF ANTS

A Thesis

by

DANA N. JABER

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Approved by:

Chair of Committee,	Costas N. Georgiades
Committee Members,	Jean-Francois Chamberland
	Shankar Bhattacharyya
	Riccardo Bettati
Head of Department,	Costas N. Georgiades

August 2006

Major Subject: Electrical Engineering

ABSTRACT

Multi-Input Multi-Output (MIMO) Detection by a Colony of Ants. (August 2006)

Dana N. Jaber, B.E., Beirut Arab University, Beirut, Lebanon

Chair of Advisory Committee: Dr. Costas N. Georghiades

The traditional mobile radio channel has always suffered from the detrimental effects of multipath fading. The use of multiple antennae at both ends of the wireless channel has proven to be very effective in combatting fading and enhancing the channel's spectral efficiency. To exploit the benefits offered by Multi-Input Multi-Output (MIMO) systems, both the transmitter and the receiver have to be optimally designed. In this thesis, we are concerned with the problem of receiver design for MIMO systems in a spatial multiplexing scheme. The MIMO detection problem is an NP -hard combinatorial optimization problem. Solving this problem to optimality requires an exponential search over the space of all possible transmitted symbols in order to find the closest point in a Euclidean sense to the received symbols; a procedure that is infeasible for large systems. We introduce a new heuristic algorithm for the detection of a MIMO wireless system based on the Ant Colony Optimization (ACO) metaheuristic. The new algorithm, AntMIMO, has a simple architecture and achieves near maximum likelihood performance in polynomial time.

ACKNOWLEDGMENTS

I would first like to thank my advisor, Dr. Costas Georghiades, for his guidance, ideas, encouragement, understanding and support during the past two years towards the publication of this thesis. I also thank my other committee members, Dr. Erchin Serpedin, Dr. Shankar Bhattacharyya, Dr. Riccardo Bettati, and especially Dr. Jean-Francois Chamberland for graciously joining the committee on short notice. Financial support from National Instruments is gratefully acknowledged; I am thankful for all the help and technical support that was provided for coding AntMIMO in LABVIEW.

On a personal note, I would like to thank the great group of friends I met during my time at A&M: Sam Bucher, Julia Croy, Myung Jun Lee, Farhan Mansoor, Indira Mohandas, Daniel Shultz, Ryan Steed, and especially Parimal Parag for his boundless enthusiasm and encouragement and Salim El-Rouayheb, my fellow Lebanese friend, for all the discussions we had about Lebanese politics and cuisine and for answering the million questions I always had.

Finally, I couldn't be where I am at today without the unconditional love and support of my family and fiancé. I thank my parents for believing in me and giving me the courage to spread my wings and fulfill my dreams, and I thank my fiancé, Mazen Diab, for being there for me during the roughest times. Mazen, I wouldn't have made it without you.

TABLE OF CONTENTS

CHAPTER		Page
I	INTRODUCTION	1
	A. Motivation	1
	B. System Model	2
	C. Thesis Outline	4
II	OVERVIEW OF MIMO SYSTEMS	6
	A. Capacity of MIMO Systems	6
	1. Ergodic Capacity	7
	2. Outage Capacity	9
	B. The BLAST System	11
	C. Space-Time Codes	12
	1. Space-Time Block Codes	13
	2. Space-Time Trellis Codes	14
III	MIMO RECEIVER DESIGN	16
	A. The Maximum-Likelihood Detector	16
	B. The Sphere Decoder	18
	1. The Zero-forcing Receiver	21
	2. The Linear MMSE Receiver	22
	3. The BLAST Receiver:	22
	4. The SDP Receiver	24
IV	INTRODUCTION TO ANT COLONY OPTIMIZATION	26
	A. Heuristics and Metaheuristics	26
	B. Ants and Natural Optimization	29
	C. Ant Colony Optimization Metaheuristic	31
	1. Problem Representation	32
	2. The Metaheuristic	32
	3. Ant System (AS)	34
	4. ACO Variants	36
	a. Ant Colony System (ACS)	36
	b. Max-Min Ant System (<i>MMAS</i>)	37
	c. Approximate Nondeterministic Tree Search (ANTS)	38
	D. Performance of ACO Algorithms	39

CHAPTER		Page
	1. ACO Parameters	39
	2. ACO and Local Search	40
	3. Convergence Proofs	40
V	SOLVING THE MIMO DETECTION PROBLEM USING ANT COLONY OPTIMIZATION	41
	A. Problem Representation	41
	B. AntMIMO	43
	1. Algorithm Parameters	48
	2. Algorithm Performance	51
	a. Simulation Results	52
	b. Effect of Parameter Settings on AntMIMO's Performance	53
	c. Complexity of the AntMIMO Algorithm	60
	d. Convergence of the AntMIMO Algorithm	61
	e. Fitness Landscape Analysis of Problem P_{MIMO}	61
VI	SUMMARY	67
	A. Future Work	68
	REFERENCES	69
	APPENDIX A	75
	VITA	80

LIST OF FIGURES

FIGURE		Page
1	Schematic representation of a MIMO system.	2
2	Ergodic capacity of a MIMO system with CSIR and no CSIT.	8
3	Asymptotic behavior of the ergodic capacity of a MIMO system.	9
4	Ergodic capacity of a 4×4 MIMO system with and without water filling. .	10
5	Complementary Cumulative Distribution Function (CCDF) plot of the outage capacity of MIMO systems.	11
6	Schematic representation of the BLAST system.	12
7	Schematic representation of the Alamouti space-time encoder.	13
8	Performance of the SD algorithm for a 4-QAM 2×2 MIMO system.	18
9	Geometric representation of the sphere decoding algorithm.	19
10	Tree representation of the sphere decoding algorithm.	20
11	Experiment setup.	30
12	Foraging behavior of ants at the beginning of the search.	30
13	Foraging behavior of ants after some time t has elapsed.	31
14	The ACO metaheuristic in pseudo-code.	33
15	Schematic representation of an $n \times n$ MIMO system in a SM configu- ration using the AntMIMO algorithm at the receiver.	41
16	Tree representation of the P_{MIMO} problem for a 4-QAM, 2×2 MIMO system.	43
17	A colony of five ants at the first level of the construction tree.	44
18	AntMIMO at the end of a construction phase.	45

FIGURE		Page
19	CDF plot of the cost function $f(x)$ normalized by σ^2 for a 4×4 MIMO system.	49
20	AntMIMO vs. Maximum-Likelihood BER plots.	53
21	AntMIMO vs. SD, ZF, and MMSE decoders for a 4×4 MIMO system with 4-QAM.	54
22	Influence of the ant colony's size on the performance of AntMIMO for a 4×4 MIMO system with 4-QAM constellation.	55
23	Influence of the evaporation rate on the performance of AntMIMO for a 4×4 MIMO system with 4-QAM constellation.	56
24	Performance of iteration-best update rule vs. best-so-far update rule for a 4×4 MIMO system using 4-QAM constellation.	57
25	Performance of a restart mechanism with $ct = 2$ vs. no restart mechanism for a 4×4 MIMO system using 4-QAM constellation.	58
26	Performance of AntMIMO plus local search vs. AntMIMO minus local search for a 4×4 MIMO system using 4-QAM constellation.	59
27	Pheromone array stem plot at $t = 4$ for a 4×4 MIMO system with 4-QAM constellation.	60
28	Pheromone array stem plot at $t = 12$ for a 4×4 MIMO system with 4-QAM constellation.	64
29	Pheromone array stem plot at $t = 25$ for a 4×4 MIMO system with 4-QAM constellation.	65
30	Fitness distance scatter plot with $\varrho = 0.941$	66
31	Schematic representation of the AntMIMO VI.	75
32	Front panel of the AntMIMO VI.	76
33	Block diagram of the AntMIMO VI.	77
34	Schematic representation of the initialization VI.	78
35	Schematic representation of the construction VI.	78

FIGURE		Page
36	Schematic representation of the ant-choose-pr subVI.	78
37	Schematic representation of the cost evaluation VI.	79
38	Schematic representation of the local search VI.	79
39	Schematic representation of the pheromone update VI.	79

CHAPTER I

INTRODUCTION

Recent trends in wireless mobile communications reflect an increasing demand for wireless multimedia services as well as an increasing number of subscribers. Since current air interfaces are incapable of supporting the required high data rates and high quality of service typically associated with broadband services, system designers have been looking into new techniques that would improve average and peak bit rates, latency, service coverage, and most importantly, spectral efficiency and system capacity. One of the most promising techniques is the use of multiple antennae at both the transmitting and receiving sides of the radio channel.

A. Motivation

The study of Multi-Input Multi-Output wireless systems has been an active research area for the past decade. Information theoretic results as well as computer simulations demonstrate the advantages of using multiple antennae systems as opposed to the single antenna system scenario. Great benefits can be achieved in terms of improved data rates and enhanced link reliability.

Traditionally, transmission over a wireless medium has always faced a number of obstacles such as path loss and interference from nearby users, but the performance of the radio channel has been mainly governed by fading [1], a phenomenon where the received signal exhibits fluctuations in the signal level due to the presence of scatterers between the transmitter and the receiver. When a channel experiences a deep fade, data is lost and the channel is rendered temporarily useless. The use of multiple antennae helps mitigate

The journal model is *IEEE Transactions on Automatic Control*.

the impairing effects of fading; in fact, the MIMO model exploits the very presence of the rich scattering environment to provide diversity and improve the performance of the wireless channel.

To harness the potential diversity and multiplexing gains of the multiple antennae system, the transmitter and the receiver should be optimally designed. For the rest of this work, we will concentrate on receiver design for the spatial multiplexing scheme, where we have t different uncoded data symbols transmitted from t antennae during a single use of a wireless channel employing t transmit and r receive antennae.

B. System Model

The general scheme of data transmission in a MIMO wireless communication system with t transmit and r receive antennae is illustrated in Fig. 1.

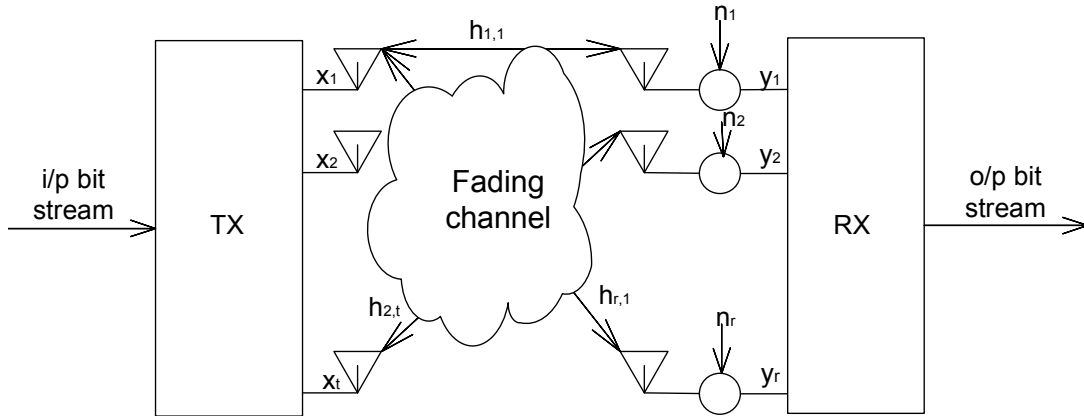


Fig. 1. Schematic representation of a MIMO system.

The information source generates a sequence of input data bits; the transmitter modulates these bits and sends the appropriate continuous-time waveform through the fading channel. The receiver attempts to recover the corrupted transmitted signal and decide upon the data bits that have been sent. For the rest of this work, we assume that the

channel is experiencing independent Rayleigh flat fading; that is, there is no interference between symbols transmitted at different time intervals, the antennae are placed far enough (typically more than $\lambda/2$) so that individual channels h_{ij} experience independent fading, and that there is no line of sight between the transmitting and receiving antennae. We also assume that the channel matrix $\tilde{\mathbf{H}}$ is perfectly known to the receiver ($\tilde{\mathbf{H}}$ can be estimated by means of sending known training symbols [2, 3]).

Mathematically, a complex baseband MIMO system is represented by the linear model:

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\tilde{\mathbf{x}} + \tilde{\mathbf{n}} \quad (1.1)$$

where

- $\tilde{\mathbf{y}}$ is an $r \times 1$ complex received vector.
- $\tilde{\mathbf{H}}$ is an $r \times t$ complex matrix representing the Rayleigh flat fading channel. Each entry $\tilde{\mathbf{h}}_{ij}$ represents the complex path gain between the j^{th} transmit and i^{th} receive antenna. Assuming the presence of a rich scattering environment, the columns of $\tilde{\mathbf{H}}$ are independent and entries $\tilde{\mathbf{h}}_{ij}$ are modeled as independent zero mean complex Gaussian random variables with unit variance.
- $\tilde{\mathbf{x}}$ is a $t \times 1$ complex transmitted vector whose elements are drawn from an M -QAM constellation.
- $\tilde{\mathbf{n}}$ is an $r \times 1$ complex noise vector whose components $\tilde{\mathbf{n}}_i$ are modeled as zero mean independent complex Gaussian random variables with variance σ^2 per real dimension.

Note that a complex vector $\tilde{\mathbf{u}}$ and a complex matrix $\tilde{\mathbf{M}}$ can be transformed into their real equivalent counterparts u and M according to :

$$u = [\Re(\tilde{\mathbf{u}})^T \Im(\tilde{\mathbf{u}})^T] \quad (1.2)$$

and

$$A = \begin{bmatrix} \Re(\tilde{\mathbf{A}})^T & \Im(\tilde{\mathbf{A}})^T \\ -\Im(\tilde{\mathbf{A}})^T & \Re(\tilde{\mathbf{A}})^T \end{bmatrix} \quad (1.3)$$

Using appropriate scaling of the transmit vector \mathbf{x} and the noise vector \mathbf{n} and the vector and matrix transformations of (1.2, 1.3), we replace the system model of (1.1) by its real-valued equivalent model,

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \quad (1.4)$$

In this model, the transmit vector x , of dimension $2t \times 1$, belongs to the set $\mathbb{Z}_{\sqrt{M}}^{2t}$, with $\mathbb{Z}_{\sqrt{M}} = \{0, \dots, \sqrt{M} - 1\}$ denoting the set of integers residues modulo \sqrt{M} .

C. Thesis Outline

In Chapter II, we summarize the information theoretic results on the capacity of MIMO systems and present some of the architectures that require the use of multiple antennae in a wireless scenario. Chapter III is concerned with the MIMO detection problem; we formally introduce Maximum-Likelihood (ML) detection for MIMO channels as an *NP*-hard problem. We survey some of the most popular techniques that have been suggested in the literature to approximate the behavior of the ML detector; we discuss the performance and complexity of such algorithms so that they can be compared to our proposed algorithm, AntMIMO.

In Chapter IV, we introduce the Ant Colony Optimization (ACO) metaheuristic, an algorithmic paradigm that was used to design the AntMIMO algorithm. The ACO metaheuristic, inspired by the behavior of real colonies of ants, is capable of solving hard combinatorial optimization problems. Chapter IV discusses the different elements of the metaheuristic and surveys some of the most famous ACO algorithms.

Chapter V presents the main contribution of this thesis. We describe in detail the AntMIMO algorithm; we show, using Monte Carlo simulations, that when used for the de-

tection of MIMO systems, AntMIMO approximates the maximum-likelihood performance with polynomial complexity.

Chapter VI summarizes the conclusions from this work and discusses areas of future research.

CHAPTER II

OVERVIEW OF MIMO SYSTEMS

This chapter presents a brief overview of the research that has been conducted in the area of multi-input multi-output (MIMO) wireless systems. In Section 2.1, we summarize some of the information theoretic results on MIMO capacity [4, 5]. These results demonstrate the significant potential gains in channel capacity that can be achieved when using multiple antennae systems. To capitalize these gains, researchers have developed some practical techniques and signal processing algorithms suitable for multi-input multi-output wireless scenarios. In Section 2.2, we describe the BLAST system [5, 6] that has been proposed to achieve high data rate wireless transmission, and in Section 2.3, we discuss space-time coding techniques, particularly space time block codes, that aim at increasing the reliability of transmission over a wireless channel.

A. Capacity of MIMO Systems

Channel capacity is the maximum data rate that a channel can support with an arbitrarily low probability of error. It is also defined as the maximum mutual information between vectors $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$, where the maximization is taken over all possible probability distributions of the random vector $\tilde{\mathbf{x}}$ [7]. Based on the system model described in Section 1.2 where the channel matrix $\tilde{\mathbf{H}}$ is random, the information rate associated with the MIMO channel is also random. Thus, we will discuss two notions of capacity: Ergodic Capacity and Outage Capacity. We would like to point out that, in the following, we will present only those results pertaining to a single user MIMO system under the assumptions made in Section 1.2.

1. Ergodic Capacity

The ergodic capacity of a MIMO channel is the ensemble average of the information rate over the distribution of the elements of the channel matrix $\tilde{\mathbf{H}}$ [8]. When the channel is known to the receiver but not to the transmitter, the ergodic capacity is given by [4, 5]

$$C = E_{\tilde{\mathbf{H}}} \left[\log_2 \det(\mathbf{I}_r + \frac{\rho}{t} \tilde{\mathbf{H}} \tilde{\mathbf{H}}^H) \right] \quad \text{bps/Hz} \quad (2.1)$$

where $\tilde{\mathbf{H}}^H$ is the transpose-conjugate of $\tilde{\mathbf{H}}$ and ρ is the SNR at any receive antenna; it is assumed that the transmitted signal vector $\tilde{\mathbf{x}}$ is composed of t statistically independent equal power components each with a gaussian distribution.

To gain some insight on the significance of the above result, (3) can be rewritten as [4]

$$C = \sum_{i=1}^m E \left[\log_2 \left(1 + \frac{\rho}{t} \lambda_i \right) \right] \quad \text{bps/Hz} \quad (2.2)$$

where $m = \min(t, r)$ and $\lambda_1 \dots \lambda_m$ are the eigenvalues of the matrix $\tilde{\mathbf{H}} \tilde{\mathbf{H}}^H$. Here, the capacity of the MIMO channel is expressed as the sum of the capacities of m parallel SISO channels ($C_{SISO} = \log_2(1 + \rho|h|^2)$ bps/Hz), each having power gain λ_i and signal-to-noise ratio equal to ρ . Fig. 2 shows the significant gains in capacity when using multiple antennae as opposed to single antenna systems.

Furthermore, it can be shown that when both t and r increase, the capacity grows linearly with $\min(t, r)$. On the other hand, if t is fixed and r is allowed to increase, the capacity increases logarithmically with r ; whereas if r is fixed and t is increased, the capacity saturates at some fixed value. Fig. 3 illustrates these asymptotic behaviors.

So far we have assumed that the channel state information is known only at the receiver. Another possible scenario is that the channel state information is known at both the transmitter and the receiver; this assumes the presence of an ideal feedback link from

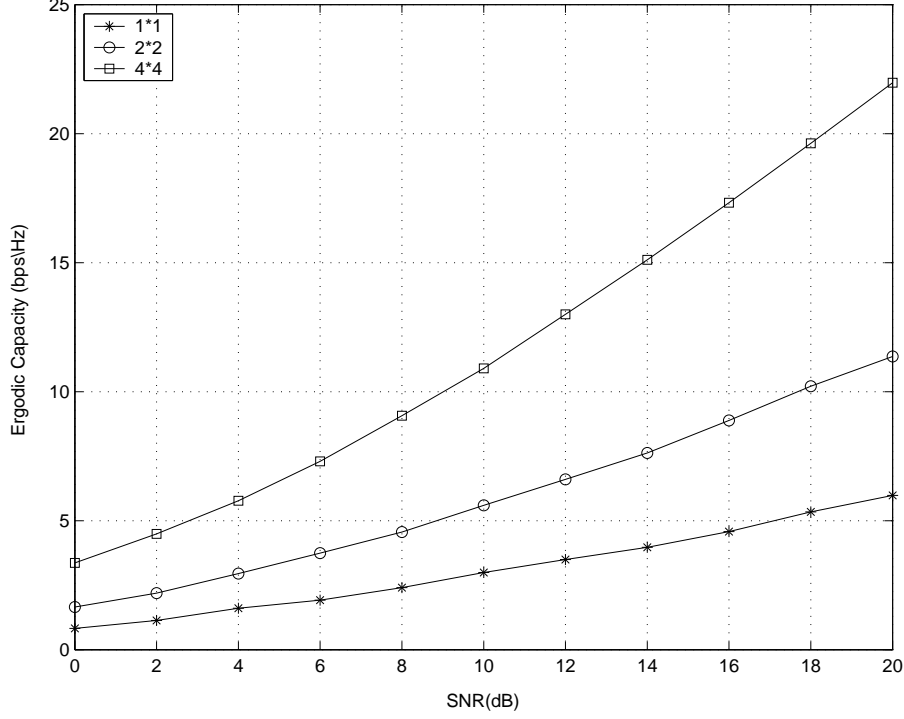


Fig. 2. Ergodic capacity of a MIMO system with CSIR and no CSIT.

the receiver to the transmitter and a very slowly fading channel for this assumption to be feasible in practice. With this knowledge of the channel, the total transmitted power can now be allocated in the most efficient way over the different transmitting antennae to achieve the highest possible bit rate; this is done using the water-filling algorithm [7]. The ergodic capacity is then given by

$$C = \mathbb{E} \left[\sum_{i=1}^t (\log_2(\nu \lambda_i))^+ \right] \quad \text{bps/Hz} \quad (2.3)$$

where λ_i is the i^{th} eigenvalue of $\tilde{\mathbf{H}}\tilde{\mathbf{H}}^{\mathbf{H}}$ and the parameter ν is chosen such that it satisfies the instantaneous power constraint $P_t = \sum_{i=1}^t (\nu - \frac{1}{\lambda_i})^+$. The notation a^+ denotes $\max(0, a)$. We expect the ergodic capacity when the channel is known to the transmitter to be higher than when the channel is unknown. This is illustrated in Fig.4; we notice

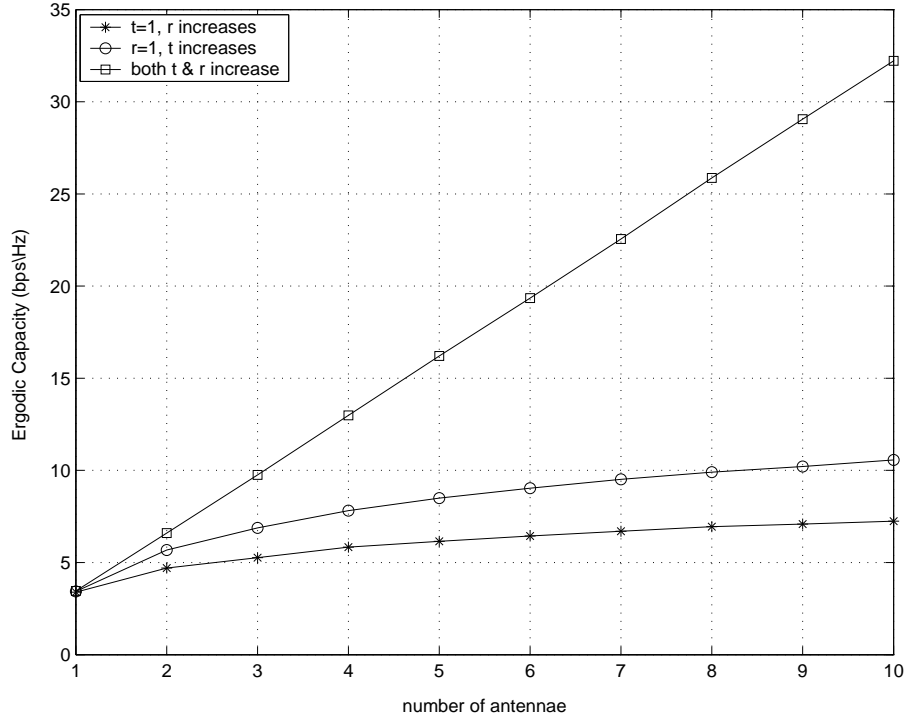


Fig. 3. Asymptotic behavior of the ergodic capacity of a MIMO system.

that the water filling capacity is indeed higher, yet at high SNR the gap between the two curves is reduced.

2. Outage Capacity

Since in a Rayleigh fading environment the channel matrix $\tilde{\mathbf{H}}$ changes randomly, the capacity is also random. One way to express the capacity of such a channel is the ergodic expression of (2.2). However, suppose that a “bad” realization of $\tilde{\mathbf{H}}$ occurs, then no matter how small the rate that we attempt to communicate at, there is a non-zero probability that this realized $\tilde{\mathbf{H}}$ is incapable of supporting this rate no matter how long we take our code length. For such a scenario, we introduce the concept of outage probability q , which is the fraction of time the capacity falls below a given threshold C_{outage} and is given by

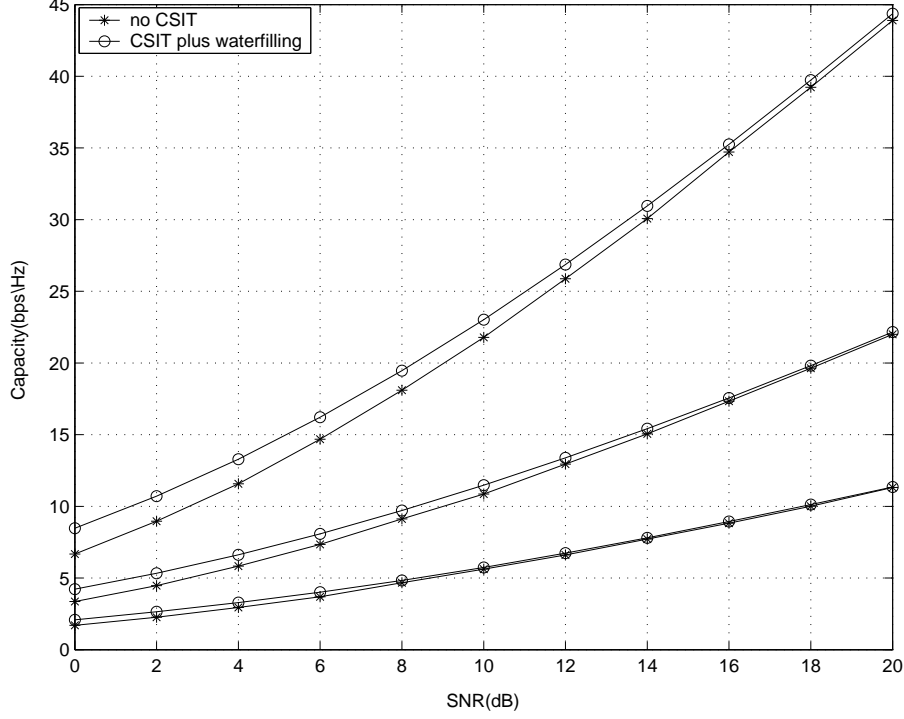


Fig. 4. Ergodic capacity of a 4×4 MIMO system with and without water filling.

$$q = Pr\{C \leq C_{outage}\} \quad (2.4)$$

where C is the instantaneous capacity given by $C = \log_2 \det(\mathbf{I}_r + \frac{\rho}{t} \tilde{\mathbf{H}} \tilde{\mathbf{H}}^H)$. A capacity of 20 bps/Hz with 1% outage probability means that a data rate of 20bps/Hz is supportable for 99% of the time. As in the case of ergodic capacity, the outage capacity increases with SNR and is higher for larger antennae configurations. For this definition of capacity, we are usually interested in the Complementary Cumulative Distribution Function (CCDF) plots of the capacity. The capacity CCDFs as a function of SNR are shown in Fig. 5, where a SISO and a 4×4 MIMO system are depicted for comparison. We see that at a 10% outage probability level, we have a significant increase in capacity for every 2 dB increase in SNR whereas for the SISO case, the increase is not even visible at low SNRs

and is only a fraction of a bit at high SNRs.

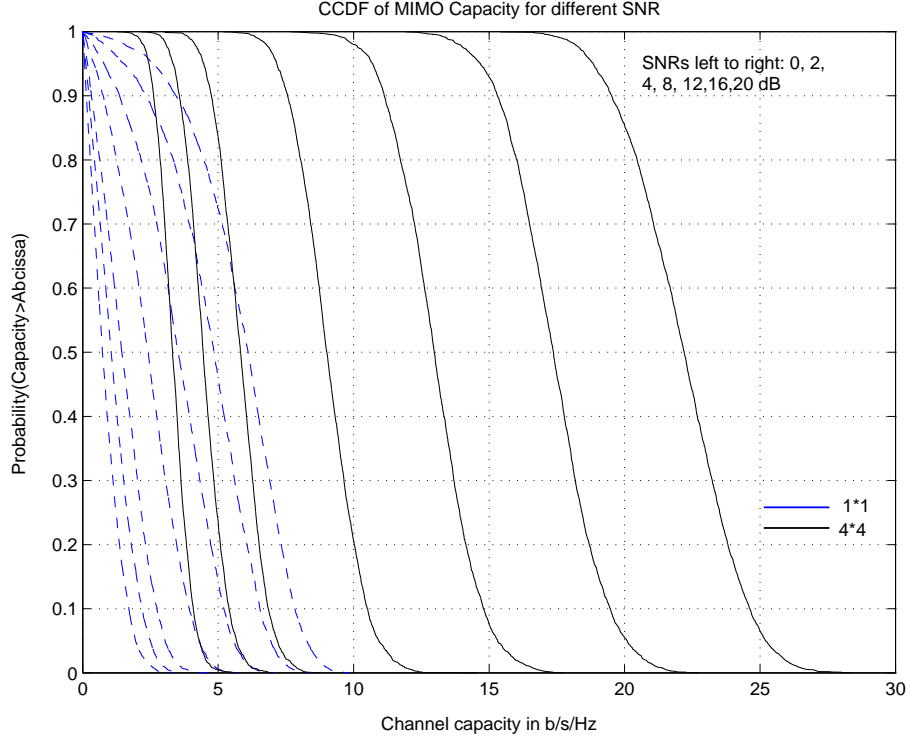


Fig. 5. Complementary Cumulative Distribution Function (CCDF) plot of the outage capacity of MIMO systems.

B. The BLAST System

The information theoretical results from the preceding section indicate the enormous increase in capacity when employing multiple antennae at both ends of the radio channel. Realizing such a potential gain, researchers at Bell-Labs developed the first MIMO architecture for high-speed wireless communications, the Bell-Labs Layered Space Time a.k.a BLAST system. In a BLAST system, the input data stream is demultiplexed into t substreams; independent bit-to-symbol mapping of each substream is performed at each of the t transmit antennae. The generated continuous-time waveforms are then simultane-

ously launched into the wireless channel overlapping in time and frequency. The signals are received by the r receive antennae as shown in Fig. 6 and signal processing at the receiver attempts to unmix the received signals and recover the transmitted data. Measurement campaigns of the BLAST system showed the great increase in spectral efficiency at reasonable SNR and BER vs. a SISO system [9]. Several variants of the original Blast system have been developed with various detection techniques [9]; these techniques will be covered in Chapter II.

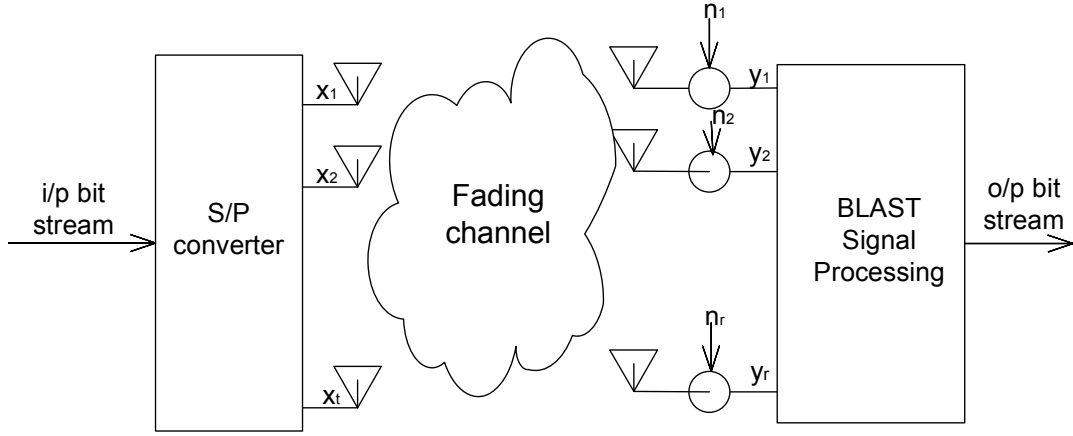


Fig. 6. Schematic representation of the BLAST system.

C. Space-Time Codes

Earlier, we have suggested the use of multiple antennae as an effective means to combat fading. This is feasible through the concept of diversity. Diversity provides redundant replicas of the transmitted message such that if this message has a probability p of being received erroneously, this probability becomes p^d when using a system with diversity order d . The three main forms of diversity typically used in wireless systems are: temporal diversity, frequency diversity, and spatial (antenna) diversity. In MIMO systems, we are interested in a combined spatial-temporal diversity, often known as *transmit* diversity. A

$t \times r$ MIMO system offers a tr -diversity gain i.e tr possible means of getting the message across the channel correctly.

In order to achieve this diversity gain, Space-Time Codes (STC) are typically used. Space-time codes introduce redundancy across space and time by spreading the information message across the multiple transmit antennae achieving spatial diversity and over multiple signaling intervals achieving temporal diversity. The amount of redundancy introduced by a space-time code is quantified by its rate, and the effectiveness of the redundancy is quantified by the diversity order. There are two main types of STCs, namely Space-Time Block Codes (STBC) and Space-Time Trellis Codes (STTC).

1. Space-Time Block Codes

As its name suggests, the space-time block encoder operates on a block of input symbols producing a code matrix. We will illustrate this encoding operation by discussing the pioneering Alamouti scheme [10], which is one of the simplest and most elegant space-time codes. As shown in Fig. 7, the information bits are first modulated according to

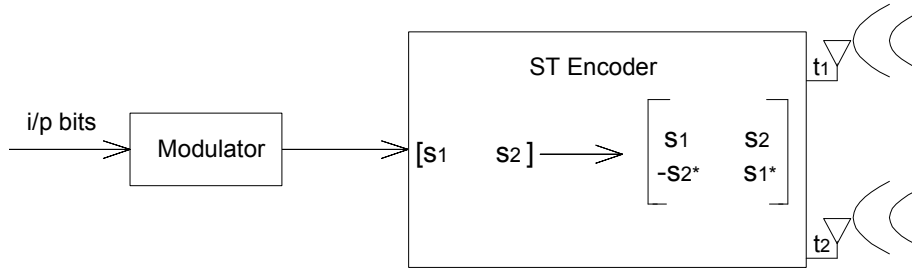


Fig. 7. Schematic representation of the Alamouti space-time encoder.

some M -ary modulation scheme. The encoder takes a block of two modulated symbols,

\mathbf{s}_1 and \mathbf{s}_2 , and produces the corresponding code matrix S

$$[\mathbf{s}_1 \quad \mathbf{s}_2] \rightarrow S = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 \\ -\mathbf{s}_2^* & \mathbf{s}_1^* \end{bmatrix} \quad (2.5)$$

The rows of S represent transmission over space and its columns represent transmission over time; thus, in the first signalling interval, \mathbf{s}_1 is transmitted from transmit antenna 1 and \mathbf{s}_2 is transmitted from transmit antenna 2; in the next signalling interval, $-\mathbf{s}_2^*$ is transmitted from antenna 1 and \mathbf{s}_1^* from antenna 2. A close examination shows that matrix S is orthogonal; this results in a diversity gain of order $2r$ for a $2 \times r$ MIMO system with Alamouti encoding. Moreover, the orthogonality of the code matrix greatly simplifies the decoder design on the receiving end of the channel; Maximum-Likelihood performance is obtained with linear complexity. The simplicity and efficiency of the Alamouti scheme inspired its generalization to more than two transmit antennae; the new codes, referred to in the literature as space-time block codes, maintain the orthogonality property of the code matrix thus achieving full diversity with a very simple decoding scheme [11].

2. Space-Time Trellis Codes

Space-time block codes do not generally provide coding gain, unless concatenated with an outer code. Space-time trellis codes, on the other hand, provide coding gain that depends on the complexity of the code, in addition to providing diversity gain. STTCs were first introduced by Tarokh, Seshadri, and Calderbank in 1998 [12]; they are an extension of trellis coded modulation. A STTC encoder maps binary data into modulation symbols according to a trellis diagram, then spreads the coded symbols over space and time as in STBCs. Their disadvantage is that they are extremely hard to design and generally require high complexity encoders and decoders; STTCs decoding is usually implemented using a Viterbi algorithm whose complexity grows exponentially with the number of states

in the trellis [12].

CHAPTER III

MIMO RECEIVER DESIGN

In the previous chapter, we have shown how the use of multiple antennae can greatly increase the spectral efficiency and/or the reliability of a mobile radio channel without increasing system bandwidth or transmission power. Yet, the design of commercially feasible MIMO systems is not as simple as adding the extra antennae or choosing the right modulation and coding schemes. The catch lies in designing simple and efficient receivers that can harness the benefits of the MIMO architecture without draining the mobile receiver's battery power or taking a long time to decode the transmitted symbols. In this chapter, we will introduce the Maximum-Likelihood detection problem and we will review some of the detection architectures that have been associated with MIMO systems in the uncoded spatial multiplexing context and comment on their complexity and performance as opposed to the optimum Maximum-Likelihood detector.

A. The Maximum-Likelihood Detector

The optimum receiver that is capable of detecting the transmitted data vector \mathbf{x} while minimizing the probability of making an erroneous decision, assuming equally likely uncoded transmit symbols, is the Maximum-Likelihood (ML) detector [13]. When the system model is given by $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$, the ML detector maximizes the likelihood that \mathbf{y} was received given that \mathbf{x} was sent,

$$\max_{\mathbf{x} \in \mathbb{Z}^{2t}_{\sqrt{M}}} p_{\mathbf{y}/\mathbf{x}}(\mathbf{y}/\mathbf{x}) \quad (3.1)$$

which, in the presence of AWGN, is equivalent to solving the following non-linear optimization problem:

$$\min_{\mathbf{x} \in \mathbb{Z}_{\sqrt{M}}^{2t}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \quad (3.2)$$

The optimization is performed over the space of all possible vectors \mathbf{x} . Since the search space is discrete with \mathbf{x} having integer components, this problem is posed in the literature as an integer least-squares optimization problem [14], and it belongs to the class of nondeterministic polynomial-time hard, *NP*-hard, combinatorial optimization problems [15, 16].

A combinatorial optimization (CO) problem involves finding values for discrete variables such that the optimal solution with respect to a given objective function is found [17]. A straight forward approach to the solution of a CO problem would be exhaustive search, i.e. the enumeration of all possible solutions and choosing the one that minimizes the objective function. A naive implementation of this search strategy results in a prohibitive complexity, as the number of candidate solutions increases exponentially with the problem size. For a $t \times r$ MIMO system with symbols $\tilde{\mathbf{x}}$ drawn from M -QAM constellation, the search space of size M^t grows exponentially with t .

Two classes of algorithms are available for the solution of combinatorial optimization problems: exact and approximate algorithms. Exact algorithms are guaranteed to find the optimal solution and to prove its optimality for every finite size instance of a combinatorial optimization problem within an instance-dependent run time; however, for *NP*-hard problems, exact algorithms have an exponential worst-case complexity, and they generally suffer from a strong rise in computation time when the problem size increases. Approximate algorithms, on the other hand, trade optimality for efficiency; they exploit some problem-specific knowledge to produce very good solutions at relatively low computational cost without being able to guarantee the optimality of the produced solutions.

In the following, we will briefly review some of the exact and approximate algorithms that have been used to solve the MIMO detection problem.

B. The Sphere Decoder

In the last decade, the sphere decoding (SD) algorithm of Fincke and Pohst [18] has been introduced as an *exact* algorithm [19, 20] for the MIMO detection problem with ML-performance, Fig. 8.

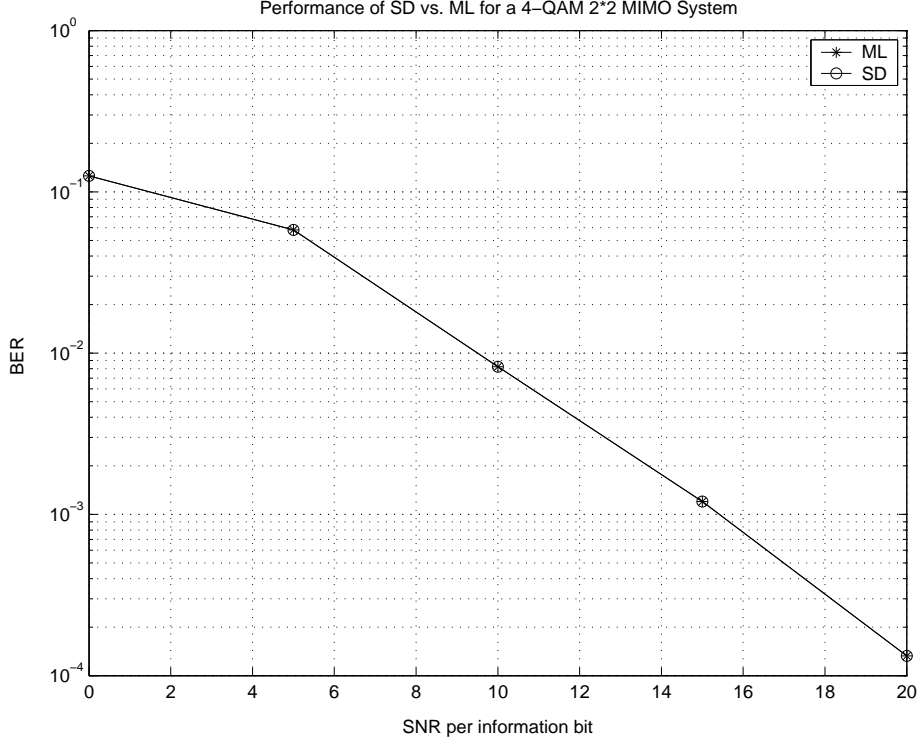


Fig. 8. Performance of the SD algorithm for a 4-QAM 2*2 MIMO system.

The transmitted data vectors are interpreted as points on a t -dimensional integer-grid rectangular lattice. The MIMO channel matrix \mathbf{H} is considered a *lattice-generating matrix* and the r -dimensional vector $\mathbf{H}\mathbf{x}$ spans a skewed lattice. Therefore, given the skewed lattice $\mathbf{H}\mathbf{x}$ and the received vector \mathbf{y} , the ML problem reduces to finding the “closest” lattice point in a Euclidean sense to \mathbf{y} . The main idea behind sphere decoding is to search over only those lattice points that lie within a hypersphere of radius R around the received vector \mathbf{y} , rather than searching over the entire lattice, as illustrated in Fig.

9.

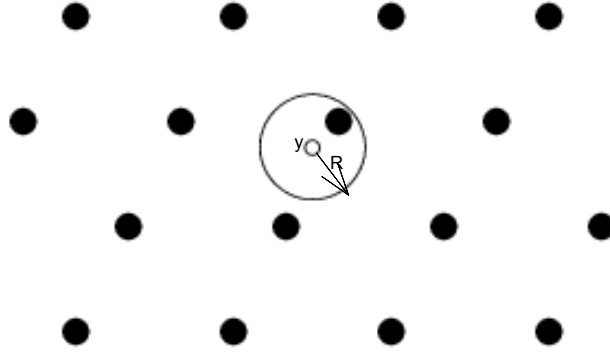


Fig. 9. Geometric representation of the sphere decoding algorithm.

Clearly, the point in the hypersphere closest to \mathbf{y} is also the closest lattice point for the whole lattice. Different variations of the SD algorithm address the issues of selecting the radius R and determining which lattice points lie inside the sphere. These two issues can dramatically improve or degrade the complexity of the algorithm.

The Sphere Decoder algorithm belongs to the family of Branch and Bound tree search algorithms [20]. Performing a QR factorization of the channel matrix \mathbf{H} exposes the inherent tree structure of the problem; thus, replacing \mathbf{H} by the product of the unitary matrix \mathbf{Q} and the upper triangular matrix \mathbf{R} in (5.2) and multiplying the whole expression by \mathbf{Q}^T yields the following equivalent problem:

$$\min_{\mathbf{x} \in \mathbf{M}^n} \|\mathbf{y}' - \mathbf{R}\mathbf{x}\|^2 \quad (3.3)$$

where $\mathbf{y}' = \mathbf{Q}^T \mathbf{y}$. Due to the upper triangular nature of \mathbf{R} , the objective function of (3.3) can be rewritten in the SD algorithm's context as

$$\sum_{j=1}^n \|y'_j - \sum_{l=j}^n r_{j,l} x_l\|^2 \leq C_0 \quad (3.4)$$

where C_0 is the squared radius of an n -dimensional sphere centered at \mathbf{y}' . A necessary

have suggested various techniques (such as lattice basis reductions [20], increasing radii [23], and pruning by means of lower bounds [24] among others) to speed up the tree search procedure of the algorithm while maintaining an acceptable performance.

While the worst case complexity of the SD algorithm is still exponential, its expected complexity has been claimed to be cubic over a certain range of rate, SNR and dimension [21]. However, recently, Jalden *et al* derived an exponential lower bound on the average complexity of the SD; thus, there will always be some problem size where an approximate polynomial time algorithm is more efficient than the sphere decoder; this is especially true when using large constellations at low SNR [25].

The rest of the techniques reviewed in this chapter belong to the class of *approximate* algorithms.

1. The Zero-forcing Receiver

This receiver solves the integer least-squares problem by removing the discreteness constraint on the components of \mathbf{x} . In this case, it is well known that the solution of the problem becomes [14], assuming \mathbf{H} is invertible and known to the receiver,

$$\hat{\mathbf{x}}_{\mathbf{ZF}} = (\mathbf{H}^H \mathbf{H})^{-1} \mathbf{H}^H \mathbf{y} = \mathbf{H}^\dagger \mathbf{y} \quad (3.5)$$

where \mathbf{H}^\dagger denotes the pseudo-inverse of \mathbf{H} and $\hat{\mathbf{x}}_{\mathbf{ZF}}$ is rounded to the nearest integer in the constellation from which \mathbf{x} is selected. The *zero – forcing* notation comes from the fact that this receiver attempts to force the cross correlation between the estimation error \mathbf{e} , $\mathbf{e} = \hat{\mathbf{x}}_{\mathbf{ZF}} - \mathbf{x}$, and the transmit vector \mathbf{x} to zero.

The ZF-receiver is a linear receiver in the sense that it behaves as a linear filter, separating the different data streams to perform independent decoding on each stream, hence eliminating multistream interference. The problem with this scheme is degraded performance due to the fact that the AWGN \mathbf{n} loses the “whiteness” property, it becomes

enhanced and correlated across the data streams. Moreover, the ZF-receiver provides only $r-t+1$ diversity order out of a maximum possible r order diversity in a $t \times r$ MIMO system [8]. On the bright side, the ZF-receiver has a polynomial complexity; the computational complexity of the algorithm is determined by the complexity of calculating the pseudo-inverse of the channel matrix \mathbf{H} . For the case when \mathbf{H} is an $n \times n$ square matrix, the complexity is of cubic order, $O(n^3)$.

2. The Linear MMSE Receiver

This receiver estimates the transmitted vector \mathbf{x} by applying the linear transformation \mathbf{G} to the received vector \mathbf{y} , such that the mean square error, $\epsilon^2 = E[||\hat{\mathbf{x}} - \mathbf{x}||^2]$, is minimized, thus

$$\hat{\mathbf{x}}_{\text{MMSE}} = \mathbf{G} \mathbf{y}; \quad (3.6)$$

where

$$\mathbf{G} = (\mathbf{H}^H \mathbf{H} + \frac{1}{\text{SNR}} \mathbf{I}_t)^{-1} \mathbf{H}^H; \quad (3.7)$$

The LMMSE receiver balances multi-stream interference mitigation and noise enhancement by minimizing the total error; it was found to be of superior performance to the ZF-receiver at low SNR, and it converges to the ZF-receiver at high SNR [26, 27]. The complexity of this receiver is dominated by the matrix inversion in (3.7); it has a cubic order complexity, $O(n^3)$, for an $n \times n$ MIMO system.

3. The BLAST Receiver:

The detection algorithm associated with the BLAST architecture is the successive cancellation (SUC) algorithm. Rather than jointly decoding all of the t transmitted symbols, this nonlinear detector decodes the first transmitted symbol by satisfying the zero forcing (ZF) or minimum mean squared error (MMSE) performance criterion while treating the

rest of the data symbols as interference; then it cancels out its contribution to obtain a reduced order integer least-squares problem with $t - 1$ unknowns. The process is repeated until all the symbols are detected. The algorithm is briefly summarized below.

1. Apply the MMSE or ZF criterion to extract the first symbol x_1 ,

$$\mathbf{z} = \tilde{\mathbf{g}}\mathbf{y};$$

where $\tilde{\mathbf{g}}$ is the first row of H^\dagger or G in (3.5) and (3.7). Slice \mathbf{z} to the nearest integer constellation to decode x_1 .

2. Assuming that the decision on x_1 is correct, subtract its contribution from the received vector \mathbf{y} . The reduced signal model becomes,

$$\mathbf{y}_{-1} = \mathbf{y} - \mathbf{h}_1\mathbf{x}_1;$$

where \mathbf{y}_{-1} is the $r \times 1$ received vector with the contribution of x_1 removed.

3. Return to step 1, decode x_2 and repeat until all the symbols in \mathbf{x} have been decoded.

The computational complexity of SUC is of $O(n^4)$ for an $n \times n$ MIMO system. In general, this algorithm performs better than the ZF or MMSE receivers, but it suffers from error propagation; its performance quickly degrades if that first symbol was incorrectly decoded. A suggested improvement is the use of ordered successive cancellation (OSUC), an algorithm associated with the Vertical-BLAST architecture. The main idea behind OSUC is that rather than selecting the symbols to be decoded in their natural order as in SUC, the symbols at the beginning of each decoding stage are ordered in terms of decreasing signal-to-interference noise ratio (SINR), and the symbol with the highest SINR is selected for decoding. The complexity of OSUC is the same as that of SUC, augmented

by the cubic complexity of the ordering operation. The computational complexity of SUC can be further reduced from $O(n^4)$ to $O(n^3)$ by using the more efficient square-root algorithms [28, 29].

4. The SDP Receiver

Another approach to solving the integer least-squares problem is via convex relaxation techniques. In such techniques, the objective function to be minimized and the corresponding constraints are expressed in an equivalent relaxed form to ensure the convexity of the new problem; the new *convexified* problem is solved using mathematical programming methods; the produced solution is then discretized to the nearest constellation integer to produce $\hat{\mathbf{x}}$.

Recently, Semi-Definite Programming (SDP) techniques have been applied to the MIMO detection problem with reasonable performance [30, 31]. Semidefinite programming refers to optimization problems that can be expressed in the form [32]

$$\begin{aligned} & \text{minimize} && Tr(\mathbf{C}\mathbf{X}) \\ & \text{subject to} && Tr(\mathbf{A}_i\mathbf{X}) = \mathbf{b}_i \quad \text{for all } i = 1, \dots, m \cdot \\ & && \mathbf{X} \succeq \mathbf{0} \end{aligned}$$

where the variable is $\mathbf{X} \in \mathbf{S}^n$, the space of real symmetric $n \times n$ matrices. The vector $\mathbf{b} \in \mathbb{R}^m$, and the matrices $\mathbf{A}_i \in \mathbf{S}^n$ and $\mathbf{C} \in \mathbf{S}^n$ are given problem parameters. The $Tr(\mathbf{C}\mathbf{X})$ notation stands for the trace of the matrix $\mathbf{C}\mathbf{X}$ and the inequality $\mathbf{X} \succeq \mathbf{0}$ means \mathbf{X} is positive semidefinite.

In the MIMO detection context, the integer-least squares problem is either formulated in a higher dimension and then the nonconvex constraints are relaxed to obtain a SDP or the semi-definite program can be derived as the Lagrangian bidual, i.e., the dual program of the dual, of the integer-least squares problem [33]. In [30], the authors convert the

integer least-squares problem into a binary quadratic minimization problem using three different relaxation models that are solvable via an SDP solver. Their approach can achieve near ML performance for M-QAM constellations with a tolerance ϵ and complexity in the order of $O(n^{5.5}M^{5.5}\log(1/\epsilon))$ for $n \times n$ MIMO system.

CHAPTER IV

INTRODUCTION TO ANT COLONY OPTIMIZATION

In this chapter, we introduce the Ant Colony Optimization (ACO) metaheuristic. We first give a brief review on heuristics, metaheuristics and the role they play in solving hard optimization problems. Section 4.2 describes the underlying biological model that inspired ACO algorithms. In Section 4.3, we formally present the ACO metaheuristic; we describe, in detail, the Ant System (AS) algorithm to illustrate the underlying mechanics of the ACO approach, and we survey some of the more recent ACO algorithms and the problems to which they have been successfully applied. Finally, we discuss some of the aspects related to the performance of these algorithms, such as convergence and incorporation of local optimization heuristics.

A. Heuristics and Metaheuristics

We have mentioned earlier that the integer-least squares problem belongs to the class of NP -hard Combinatorial Optimization (CO) problems. Formally, a CO problem P is a triple (S, f, Ω) , where S is the set of candidate solutions, f is the objective function which assigns to each candidate solution $s \in S$ a cost value $f(s)$ and Ω is a set of constraints. The solutions belonging to the set $\tilde{S} \subseteq S$ of candidate solutions that satisfy the constraints Ω are called feasible solutions. To solve a combinatorial optimization problem, we have to find a *globally optimal* feasible solution $s^* \in \tilde{S}$ such that $f(s^*) \leq f(s)$ for all $s \in \tilde{S}$ if P is a minimization problem, otherwise $f(s^*) \geq f(s)$ for all $s \in \tilde{S}$ if P is a maximization problem. In this thesis, we will focus on minimization problems.

The majority of combinatorial optimization problems of interest are *NP – Complete*; i.e there exist no known polynomial time algorithms that can find the optimal solution to these problems [34]. Heuristics are approximate algorithms used to find good, but not

necessarily optimal, solutions to hard CO problems in polynomial-time. There are two main classes of heuristic methods: *construction* methods and *local search* methods.

Construction heuristics generate solutions from scratch by iteratively adding solution components to an initially empty solution until the solution is complete. Although the order in which to add solution components can be random, typically some kind of heuristic rule is involved; for example, in greedy construction heuristics, the solution component with maximum myopic benefit as estimated by a heuristic function is chosen at each construction step. Construction heuristics are the fastest among approximate algorithms, yet they often return solutions of inferior quality when compared to the quality of the solutions returned by local search algorithms. Local search algorithms start from some initial solution and iteratively try to improve the current solution by searching, within a pre-specified neighborhood, for better solutions. A *neighborhood structure* is used to specify the solutions' neighborhood.

Definition A.1. A *neighborhood structure* is a function $N : S \rightarrow 2^S$ that assigns a set of neighbors $N(s) \subseteq S$ to every $s \in S$. $N(s)$ is called the *neighborhood of s* .

The choice of an appropriate neighborhood structure greatly affects the performance of a local search algorithm and is problem specific. A local search algorithm also requires the definition of a neighborhood examination scheme that determines how a neighborhood is searched and which neighbor solutions are accepted. In the majority of local search algorithms, either the *best-improvement* rule, which returns the neighbor solution giving the largest improvement to the objective function, or the *first-improvement* rule, which accepts the first improved solution found, are employed. The size of the neighborhood and the type of acceptance rule used determine the complexity of a local search algorithm. Moreover, the solution returned by a local search algorithm is *locally optimum*; there is no guarantee that it can be *globally optimum* [17].

Definition A.2. A *locally optimum solution* for a minimization problem (a local mini-

mum) with respect to a neighborhood structure N is a solution \hat{s} such that $\forall s \in N(\hat{s}) : f(\hat{s}) \leq f(s)$. We call \hat{s} a strictly locally minimal solution if $f(\hat{s}) < f(s) \forall s \in N(\hat{s})$.

The problem with the above mentioned heuristics is that sometimes they can get trapped in some bad local optima; to increase their effectiveness, an obvious approach would be to restart the algorithm several times with some arbitrary initial solutions - or empty solutions if constructive heuristics are used -, retaining the best local optima obtained over the run time of the algorithm; however, this procedure increases the computing time of the algorithm and there is still no guarantee that we will reach a better solution. To overcome this obstacle, we now use metaheuristics.

The term *metaheuristics* was first introduced by Glover in 1986 [35]; it derives from the composition of two Greek words: *meta* which means “beyond, in an upper level” and *heuristics* which is derived from the verb *heuriskein* which means “to find”. It is formally defined as: “A metaheuristic is a set of concepts that can be used to define heuristic methods that can be applied to a wide set of different problems. In other words, a metaheuristic can be seen as a general algorithmic framework which can be applied to different optimization problems with relatively few modifications to make them adapted to a specific problem” [35]. So, instead of blindly examining the search space, metaheuristics guide the underlying heuristics to areas of the search space containing high-quality solutions according to some mechanism that is metaheuristic specific. Among the most popular metaheuristics, we mention Simulated Annealing, Tabu Search, Evolutionary Computing and Genetic Algorithms, and Ant Colony Optimization. For comprehensive overview on metaheuristics and their applications, please refer to [36, 37]. Several Metaheuristics such as tabu search and genetic algorithms have been applied to the multiuser detection problem in Code Division Multiple Access (CDMA) scenarios with competitive performance [38], yet none, and especially not the ant colony optimization metaheuristic, has been applied to the MIMO detection problem to our knowledge.

The rest of this chapter is dedicated to the ant colony optimization metaheuristic.

B. Ants and Natural Optimization

Social insects such as termites, ants, and bees are capable of complex and intelligent group behavior [39]. This collective behavior is of interest as it emerges from apparently simple, and often indirect, interactions among members of these insect groups, known as colonies. These indirect interactions are often mediated by deliberate modifications of the environment surrounding the colonies, a process known as *stigmergy*. Ants, in particular, are dominantly semi-blind - some species are completely blind -, so most of the communication among ants or between ants and the environment is based on the use of odorous volatile chemicals, called pheromones, secreted by a gland located in the ants' abdomen. Ants are very sensitive to pheromones; they perceive them because of the receivers located in their antennae. Ants use pheromones for communication in a number of contexts such as alerting other individuals to a threat, attracting mates, recognizing individuals from the same colony, marking a colony's home range, marking trails to food sources and recruiting other ants to collect food from those sources [40]. The foraging behavior of ants was of particular interest to researchers who noticed that a colony is able to choose the shortest path between the nest and a source of food even though individual ants do not have a *global* vision of the path [41]. In a controlled experiment, Deneubourg et al [41] placed a nest of ants of the Argentine ants species *I. humilis* and a food source at the opposing ends of a simple maze consisting of two paths, one longer than the other as illustrated in Fig. 11.

When ants start exploring the space surrounding the nest in search for food, they do so in a random fashion; thus when an ant reaches a fork in the road, it chooses either one of the two paths with equal probability, as there is no initial pheromone traces placed on either paths to alert the ants to favor one over the other, Fig. 12.

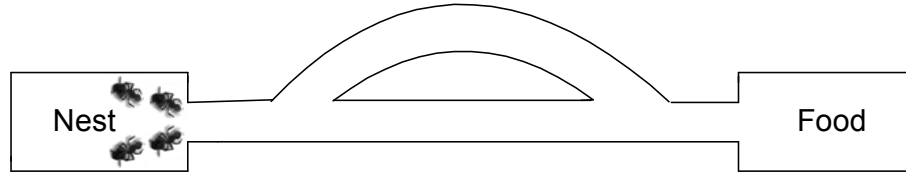


Fig. 11. Experiment setup.

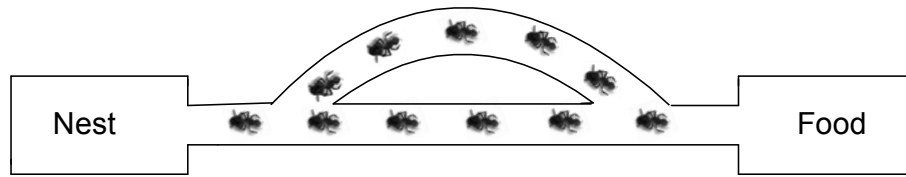


Fig. 12. Foraging behavior of ants at the beginning of the search.

When an ant reaches the food source, it retraces its path back to the nest, depositing pheromones along the way to alert other ants to the presence of food at the end of the path. Given the same period of time, an ant using the shorter path will be able to complete more trips going back and forth between the nest and the food source than an ant using the longer path; thus a larger amount of pheromones will accumulate on the shorter path. This increased amount of pheromones will bias the decision of ants when they reach the fork in the road once again and the shorter path will be chosen with a higher probability, which means a larger number of ants will traverse the path with the passage of time leading to an even larger amount of pheromones accumulating on the shorter path. At the end, almost all of the ants will choose the shorter path, due to this *positive feedback* process, Fig. 13.

The ability of ant colonies to naturally solve optimization problems, such as selecting the shortest path, inspired Dorigo, Maniezzo and Colorni to build the first ACO algorithm mimicking ants' behavior to solve hard combinatorial optimization problems . This

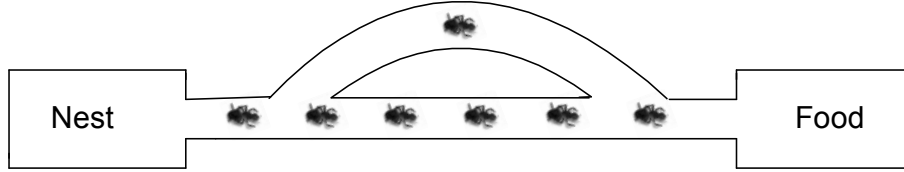


Fig. 13. Foraging behavior of ants after some time t has elapsed.

algorithm, known as Ant System [42], was the progenitor to a number of ant-inspired algorithms which were formally grouped within one general framework called the Ant Colony Optimization metaheuristic [43].

C. Ant Colony Optimization Metaheuristic

In analogy to the biological example, ACO is based on the indirect communication of a colony of artificial ants mediated by artificial pheromone trails. An ant is a simple computational agent, which probabilistically builds a solution by iteratively adding solution components to partial solutions by taking into account (i) a priori available heuristic information on the problem instance being solved and (ii) artificial pheromone trails which change dynamically at run-time to reflect the ants' collective search experience [17]. The pheromone trails and the heuristic information serve as a guide for the ants to concentrate their search in regions of the search space containing high quality solutions. Thus, ACO employs a constructive heuristic where solution components are added stochastically, rather than deterministically. It also provides a means from escaping local minima by repeatedly letting several ants construct solutions and communicate their search experience by modifying the pheromone trail, which is a dynamic memory structure available to all ants. The collective behavior emerging from the interaction of different ants proved to be effective in solving combinatorial optimization problems.

1. Problem Representation

Following the notation of [17], a CO problem $P = (S, f, \Omega)$, in an ACO context, is mapped on a problem that can be characterized by the following list of items:

- A finite set $C = \{c_1, c_2, \dots, c_N\}$ of *components* is given.
- The *states* of the problem are defined in terms of sequences $x = \langle c_i, c_j, \dots, c_h, \dots \rangle$ of finite length over the elements of C . The set of all possible states is denoted by χ . The length of a sequence x , that is the number of components in the sequence, is expressed by $|x|$. The maximum length of a sequence is bounded by a positive constant $n < +\infty$.
- The set of candidate solutions S is a subset of χ (i.e., $S \subseteq \chi$).
- The finite set of *constraints* Ω defines the set of feasible states $\tilde{\chi}$, with $\tilde{\chi} \subseteq \chi$.
- A non-empty set S^* of optimal solutions, with $S^* \subseteq \tilde{\chi}$ and $S^* \subseteq S$.
- A cost $f(s)$ is associated to each candidate solution $s \in S$.

Given this representation, artificial ants build solutions by moving on the construction graph $G = (C, L)$ whose vertices are the components C and the set L fully connects the components C (elements of L are called connections). The set of constraints Ω are implemented by the construction policy of the ants as they walk on G . Moreover, either components $c_i \in C$ or connections $l_{ij} \in L$ can have pheromone values τ and heuristic information η associated with them. These values are used by the ants to make probabilistic decisions on how to move on the construction graph.

2. The Metaheuristic

Fig. 14 presents the ACO metaheuristic in pseudo-code; all of the ACO algorithms can be described by the **ScheduleActivities** construct and its three procedures: **Construc-**

tAntsSolutions, UpdatePheromones,and DaemonActions.

```

procedure ACOMetaheuristic
  ScheduleActivities
    ConstructAntsSolutions
    UpdatePheromones
    DaemonActions      % optional
  endScheduleActivities
end-procedure

```

Fig. 14. The ACO metaheuristic in pseudo-code.

ConstructAntsSolutions allows a colony of ants to iteratively build solutions to the problem by moving through nodes of the construction graph G while applying a stochastic local decision policy. Moreover, each ant keeps in memory the path it traversed; when an ant finishes constructing a solution, it evaluates the cost of this solution in order to determine how much pheromones need to be deposited by the **UpdatePheromones** procedure to communicate its search experience to the other ants.

UpdatePheromones is the process responsible for modifying the pheromone trails; the trail values increase when ants deposit pheromones on the components or connections they used to build their solutions, but prior to that, trail values are decreased by means of evaporation. Evaporation plays an important role by regulating the amount of pheromones present on a certain path, favoring the possibility of exploring other regions of the search space and preventing the algorithm from converging rapidly to suboptimum solutions.

Recall that ants have a local, rather than global, vision of their environment. So, sometimes it is beneficial to use the **DaemonActions** procedure to implement centralized

actions which cannot be performed by single ants. Such daemon actions may include the use of a local search algorithm to improve the quality of the ants' solutions; it may also include a mechanism to reward/penalize the best/worst solution by either increasing or decreasing the corresponding pheromone trail values.

Finally, the **ScheduleActivities** construct does not specify how the three procedures are scheduled and synchronized; thus the designer has the freedom to specify how these procedures should interact and to tailor them according to the specific problem to be solved.

Next, we will see how ACO algorithms, following the ACO metaheuristic outlined above, work. We will describe the Ant System algorithm and briefly mention some of the more recent variants.

3. Ant System (AS)

As mentioned before, Ant System was the first example of an ACO algorithm to be proposed in the literature. It consisted of the procedures **ConstructAntsSolutions** and **UpdatePheromones**; no **DaemonActions** were employed. Thus, given a combinatorial optimization problem $P = (S, f, \Omega)$ with its corresponding construction graph $G = (C, L)$, AS outputs a feasible solution $\hat{s} \in S$ and $\hat{s} \in \tilde{\chi}$. Let $s_p^k = \langle c_l, c_h, \dots, c_j \rangle$ denote the k^{th} ant's partially constructed solution as it walks on construction graph G . The algorithm goes as follows, assuming pheromones, τ_{ij} , and heuristic information, η_{ij} , are associated with connections, l_{ij} , without loss in generality:

Algorithm 4.1: Ant System (AS)

input: a CO problem instance $P = (S, f, \Omega)$

Initialize τ_{ij} and $\eta_{ij} \quad \forall (i, j)$

while termination conditions not met **do**

for each ant $k = 1, \dots, m$ **do**

repeat

 choose in probability the next component c_i to add to s_p^k by means of (4.1)

 append the chosen component c_i to the k^{th} ant's list of visited nodes.

until ant k has completed its solution.

 compute the amount of pheromones $\Delta\tau_{ij}^k$ to be deposited on all connections l_{ij}

 ant k traversed when building its solution.

end for

 update pheromone values τ_{ij} on all connections l_{ij} of $G = (C, L)$ by means of (4.2)

end while

output : \hat{s}

The probability with which an ant k chooses to add component c_j to its partially constructed solution $s_p^k = \langle c_h, \dots, c_i \rangle$ is given by the *random proportional* rule,

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta}, & \text{if } j \in N_i^k \\ 0 & \text{if } j \notin N_i^k \end{cases} \quad (4.1)$$

where α and β are two user-defined parameters that determine the relative influence of the pheromone value and the heuristic information on the ant's choice; N_i^k is the feasible neighborhood of an ant k , that is, the set of allowable components c_j that can be added to s_p^k .

The pheromone update rule used in AS is given by,

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k \quad \forall (i, j) \in L \quad (4.2)$$

where $\rho \in [0,1]$ is the pheromone evaporation rate and $\Delta\tau_{ij}^k$ is the amount of pheromones deposited by ant k on connection l_{ij} ,

$$\Delta\tau_{ij}^k = \begin{cases} F(s^k) & \text{if ant } k \text{ used } l_{ij} \text{ when constructing } s^k \\ 0 & \text{otherwise} \end{cases} \quad (4.3)$$

where $F : S \rightarrow \mathbb{R}^+$ is a function that satisfies $f(s) < f(s') \implies F(s) \geq F(s'), \forall s \neq s' \in S$ when P is a minimization problem. $F(\cdot)$ is commonly called the quality function.

4. ACO Variants

Since its introduction in 1992, several researchers have aimed to improve the performance of Ant System so that it can be applied to a wide range of problems with competitive results; the improved versions of AS became stand-alone algorithms in their own right. Among the best performing ACO algorithms, we mention Ant Colony System(ACS) [44], Max-Min Ant System (MMAS) [45], and ANTS [46].

a. Ant Colony System (ACS)

The ACS algorithm has been introduced by Dorigo and Gambardella to improve the performance of AS. It differs from AS in three main aspects. First, it uses a more aggressive probabilistic choice rule, called the *pseudo-random proportional* rule; thus an ant k chooses to add its next component c_j to $s_p^k = \langle c_h, \dots, c_i \rangle$, according to,

$$c_j = \begin{cases} \arg \max_{l \in N_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta & \text{if } q \leq q_0 \\ \text{Apply AS choice rule of (4.1)} & \text{otherwise} \end{cases} \quad (4.4)$$

where $q \in [0,1]$ is a uniform random variable and $q_0 \in [0,1]$ is a parameter controlling the probability with which an ant makes the best possible move *exploiting* the algorithm's knowledge of pheromone values and heuristic information; the algorithm *explores* the search space by employing the AS choice rule of (4.1) with probability $(1 - q_0)$. The second main modification is to the **UpdatePheromones** procedure. In ACS, pheromones are updated in two stages: a local stage and a global stage. During a construction phase, after an ant k traverses connection l_{ij} , it modifies the associated pheromone value τ_{ij} according to,

$$\tau_{ij} = (1 - \zeta)\tau_{ij} + \zeta\tau_0 \quad (4.5)$$

where $\zeta \in [0,1]$ is the local pheromone evaporation rule and τ_0 is the initial amount of pheromones deposited on each connection. The use of the local update rule of (4.5) favors the exploration of yet unvisited nodes while the use of the global update rule of (4.6) concentrates the search in the neighborhood of the best-so-far found solution, s^{bs} , since the start of the algorithm.

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho\Delta\tau_{ij}^{bs} \quad \forall l_{ij} \text{ traversed while building } s^{bs} \quad (4.6)$$

Finally, ACS is the first ACO algorithm to incorporate a local optimization heuristic to improve the quality of the ants' solutions, as well as to use candidate lists to reduce the number of components to choose from at each step allowing its application to large problem instances.

b. Max-Min Ant System (*MMAS*)

MMAS is another extension of AS. The distinguishing characteristic of *MMAS* is that all pheromone values are restricted to an interval $[\tau_{min}, \tau_{max}]$ and the initial pheromone value τ_0 is set to τ_{max} ; this prevents the probability to construct a solution to fall below a certain threshold greater than 0 and favors the exploration of new regions of the search space,

especially in the earlier stages of the algorithm. Moreover, *MMAS* employs the global update rule of (4.6) sometimes substituting the best-so-far solution with the best-iteration solution or some combination of the two depending on the application. Finally, to avoid entrapment in local optima, *MMAS* re-initialize all pheromone values to τ_{max} when little change in the solutions produced over time is detected.

c. Approximate Nondeterministic Tree Search (ANTS)

ANTS is an ACO algorithm that exploits ideas from mathematical programming. In particular, ANTS computes lower bounds (upper bounds in the case of maximization problems) on the cost of completion of a partial solution to define the heuristic information η that is used by each ant during solution construction. The name ANTS derives from the fact that the algorithm can be interpreted as an approximate or incomplete probabilistic tree search with no backtracking mechanism; it was shown that it can be easily extended to a branch & bound exact algorithm. Apart from the use of lower bound, ANTS also uses a modified probability choice rule,

$$p_{ij}^k = \begin{cases} \frac{\zeta\tau_{ij} + (1-\zeta)\eta_{ij}}{\sum_{l \in N_i^k} \zeta\tau_{il} + (1-\zeta)\eta_{il}} & \text{if } j \in N_i^k \\ 0 & \text{if } j \notin N_i^k \end{cases} \quad (4.7)$$

where $\zeta \in [0,1]$ is a parameter. Another particularity of ANTS is that it has no explicit pheromone evaporation; the pheromone update rule is given by,

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (4.8)$$

where

$$\Delta\tau_{ij}^k = \begin{cases} \vartheta \left(1 - \frac{C^k - LB}{L_{avg} - LB} \right) & \forall l_{ij} \text{ traversed while building } s^k \\ 0 & \text{otherwise} \end{cases} \quad (4.9)$$

ϑ is a parameter, C^k is the cost of the current solution s^k , L_{avg} is the moving average of the cost of the last l globally constructed solutions, and LB is a lower bound on the optimal solution cost. If an ant's solution is worse than the current moving average, the pheromone values of the connections used by the ant are decreased; if the ant's solution is better, they are increased. Moreover, if $C^k = LB$, then the algorithm is stopped, since in this case the optimum solution is found.

ACO algorithms have been successfully applied to hard combinatorial optimization problems such as the Traveling Salesman Problem (TSP), the Quadratic Assignment Problem (QAP), and Graph Coloring problem among others. ACO algorithms also exist for network routing and for solving continuous optimization problems. For a comprehensive list of ACO applications, please refer to [47].

D. Performance of ACO Algorithms

In this section, we discuss some of the aspects concerning the performance of ACO algorithms.

1. ACO Parameters

In a typical ACO algorithm, the designer is faced with the problem of selecting values for a relatively large set of parameters such as initial pheromone value τ_0 , the number of ants m , and the evaporation factor ρ among others. In most cases, the different parameter values are tailored according to the application at hand, often by running experiments and selecting the set of values that produced the best results which is a tedious and time consuming procedure. In a recent paper [48], the authors used concepts from genetic algorithms in order to evolve solution parameters during the running time of the ACS algorithm, but the procedure was computationally expensive. In another approach [49], mechanisms from ant colony optimization were incorporated within an ACS algorithm to

tune the different parameters; the algorithm in this instance is solving the optimization problem at hand in addition to optimizing the parameter set, which further burdens the algorithm; however, it was found that the use of such a strategy yielded better performing algorithms, in terms of objective function values, than when using hand tuned parameters.

2. ACO and Local Search

Local search heuristics are used to improve the solutions returned by ACO algorithms; they can be easily incorporated within the **DaemonActions** procedure of the ACO metaheuristic. ACO algorithms were found to be more effective when they are *hybridized* with local search heuristics. This is due to the fact that ACO's solution construction uses a different neighborhood than local search, so the probability that local search improves a solution constructed by an ant is quite high.

3. Convergence Proofs

For a basic random search algorithm, it is guaranteed that the optimal solution will be found, but for metaheuristics such as the ACO metaheuristic, the random search procedure is biased and convergence is *not* guaranteed. In the ACO literature, there have been some convergence proofs that were derived using simplified versions of actual ACO algorithms, but no general results currently exist. The first of such proofs was proposed by Gutjahr [50] who proved the convergence in probability of a theoretical ACO variant called Graph Based Ant System (GBAS) under certain conditions. In [17], the authors distinguish between convergence in value and convergence in solution: the former refers to the case when the algorithm outputs the optimal solution at least once while the latter refers to the case when the algorithm outputs the optimal solution over and over. Asymptotic convergence in value was proved for the ACS and MMAS algorithms.

CHAPTER V

SOLVING THE MIMO DETECTION PROBLEM USING ANT COLONY OPTIMIZATION

In this chapter, we apply the Ant Colony Optimization metaheuristic to the MIMO detection problem; we show that when using ACO algorithms for MIMO detection near Maximum-Likelihood performance is achievable in polynomial-time. In Section 5.1, we pose the integer-least squares problem as a combinatorial optimization problem and represent it on a construction graph. Section 5.2 introduces the AntMIMO algorithm and discusses various aspects of the algorithm such as parameter selection, computational complexity, and performance when applied to MIMO systems.

A. Problem Representation

Given an $n \times n$ MIMO system in a spatial multiplexing mode with transmit symbols selected from a complex M -QAM constellation, we want the receiver to correctly detect the t transmitted messages using the AntMIMO algorithm.

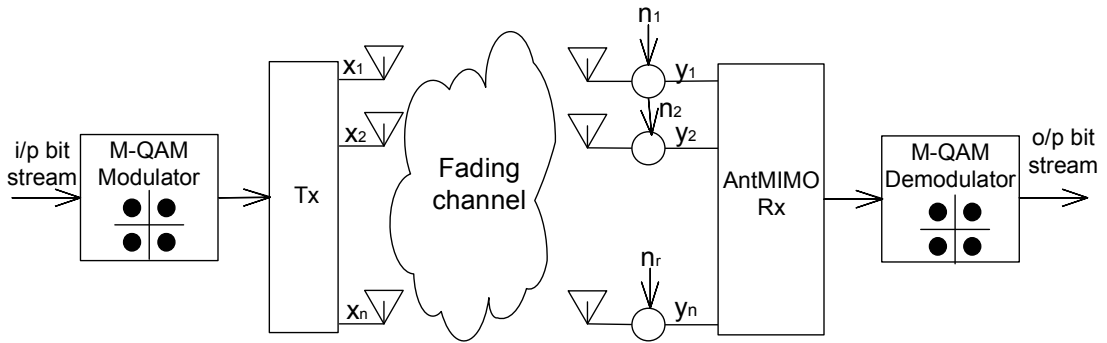


Fig. 15. Schematic representation of an $n \times n$ MIMO system in a SM configuration using the AntMIMO algorithm at the receiver.

The complex $n \times n$ MIMO system of Fig. 15 can be represented by the real system

model of Section 1.2,

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}; \quad (5.1)$$

where $\mathbf{y} \in \mathbb{R}^{2n}$, $\mathbf{H} \in \mathbb{R}^{2n \times 2n}$ with $h_{ij} \sim N(0, 0.5)$, $\mathbf{x} \in \mathbb{Z}_{\sqrt{M}}^{2n}$, and $\mathbf{n} \in \mathbb{R}^{2n}$ with $n_i \sim N(0, \sigma^2)$. The MIMO detection problem, as explained in Chapter III, is equivalent to solving the integer least-squares minimization problem

$$\min_{\mathbf{x} \in \mathbb{Z}_{\sqrt{M}}^{2n}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \quad (5.2)$$

Following the notation of Sections 4.2 and 4.3, the integer-least squares minimization problem of (5.2) is represented by the combinatorial optimization problem $P_{MIMO} = (S, f, \Omega)$ where S is the space of all possible transmit vectors \mathbf{x} , the objective function f to be minimized is the Euclidean distance $\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2$, and the set of constraints Ω impose the integrality constraint on the elements of \mathbf{x} , $x_i \in \mathbb{Z}_{\sqrt{M}}$ where $\mathbb{Z}_{\sqrt{M}} = \{0, 1, \dots, \sqrt{M}\}$.

We associate with the CO problem P_{MIMO} a complete tree $T = (C, E)$ characterized by:

- A dummy node R serving as root.
- A height $h = 2n$ for an $n \times n$ MIMO system.
- The finite set $C = \{c_{1,L_0}, c_{2,L_1}, \dots, c_{N,L_{\sqrt{M}}}\}$ represent the labeled nodes of the tree, where label L_i associated with node c_{j,L_i} is selected from the set $\mathbb{Z}_{\sqrt{M}}$.
- The set E represent the edges connecting parent nodes to children nodes.
- Each parent node has \sqrt{M} children when transmit symbols are selected from an M -QAM constellation.
- A path $w = \langle c_{i,L_j}, \dots, c_{h,L_k} \rangle$ of length $2n$ that starts at the first level of the tree where $h = 1$ and proceeds down the tree selecting a single node at each level

till a leaf node is reached is representative of the feasible solution $\hat{x} = (j, \dots, k)$ to problem P_{MIMO} .

Fig. 16 shows the complete tree associated with a 2×2 MIMO system utilizing a 4-QAM constellation. Path $w = \langle c_{2,1}, c_{5,0}, c_{12,1}, c_{25,0} \rangle$, for instance, is equivalent to feasible solution $\hat{\mathbf{x}} = (1, 0, 1, 0)$.

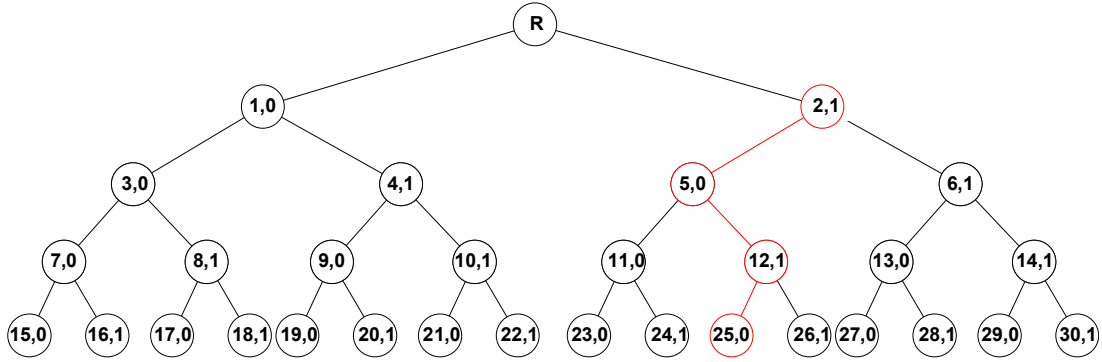


Fig. 16. Tree representation of the P_{MIMO} problem for a 4-QAM, 2×2 MIMO system.

Using this representation, the Ant Colony Optimization metaheuristic can be easily adapted to solve problem P_{MIMO} . In the following, we give a detailed description of the new algorithm, AntMIMO.

B. AntMIMO

Following the Ant Colony Optimization paradigm, AntMIMO has three distinctive procedures: `ConstructAntsSolutions`, `ApplyLocalSearch`, and `UpdatePheromones`.

1. **ConstructAntsSolutions:** A number of m ants are placed randomly at the first level of the tree, as illustrated in Fig. 17. The m ants iteratively construct solutions to problem P_{MIMO} as they walk down the tree in parallel. An ant k currently at parent

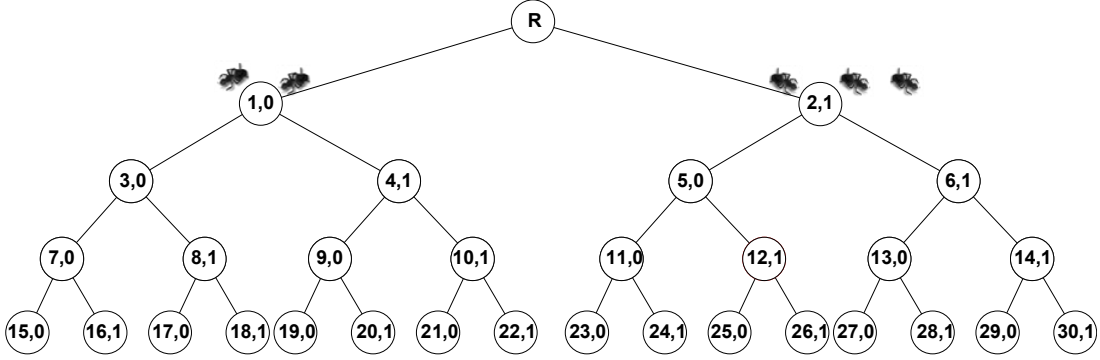


Fig. 17. A colony of five ants at the first level of the construction tree.

node c_i^1 chooses to move into one of the \sqrt{M} children nodes c_j according to the random proportional rule,

$$p_{ij}^k = \frac{\tau_{ij}}{\sum_{l \in N_i^k} \tau_{il}} \quad (5.3)$$

where τ_{ij} is the pheromone level present on the edge e_{ij} connecting node c_i to node c_j and the set N_i^k is the set all the children of node c_i . When an ant reaches a leaf node, the construction phase ends, and the ant evaluates the cost of the constructed solution $f(x^k)$, Fig. 18. Note that an ant is capable of storing a list of the nodes it visited while constructing x^k . Such a list allows the ant to retrace the path it traversed and deposit the appropriate amount of pheromones along its edges to reflect its search experience .

2. **ApplyLocalSearch:** The solutions constructed by the m ants are carried to their local optima by means of a 1-opt local search algorithm characterized by a one-flip neighborhood [51]. In the one-flip neighborhood, a solution x is a neighbor of a solution x' if x and x' differ in exactly one solution component. For example, if $x = (1, 1, 1, 1)$, then the one-flip neighborhood $N(x)$ includes $(0, 1, 1, 1)$, $(1, 0, 1, 1)$, $(1, 1, 0, 1)$, and

¹For simplicity of notation, we will use c_i to refer to a tree node instead of c_{i,L_j}

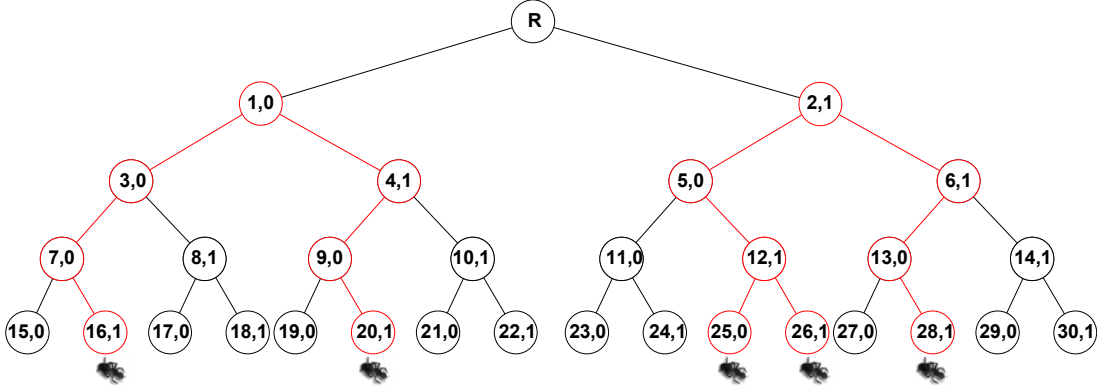


Fig. 18. AntMIMO at the end of a construction phase.

$(1, 1, 1, 0)$.

Algorithm 5.1: 1-opt Local Search

input: ant solution x^k , corresponding path w^k

$x_{LS}^k \leftarrow x^k$.

$w_{LS}^k \leftarrow w^k$.

for all $x' \in N(x)$ **do**

Calculate cost of solution x' , $f(x')$.

if $f(x') < f(x_{LS}^k)$ **then**

$x_{LS}^k \leftarrow x'$.

$w_{LS}^k \leftarrow w'$.

end if

end for

output : x_{LS}^k, w_{LS}^k

3. **UpdatePheromones** : In AntMIMO, we apply the global pheromone update rule of ACS where only the best-so-far found solution since the start of the algorithm x^{bs}

is allowed to update the pheromone trails on the edges corresponding to the path it traversed .

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho\Delta\tau^{bs} \quad \forall e_{ij} \text{ traversed while building } x^{bs} \quad (5.4)$$

where $\rho \in [0,1]$ is the evaporation rate, and $\Delta\tau^{bs}$ is defined to be

$$\Delta\tau^{bs} = 1/f(x^{bs}) \quad (5.5)$$

In addition to these three main procedures, we allow the algorithm to re-initialize its pheromone array via the procedure `ReinitializePheromones` only in the case when it returns a “bad” solution. A “bad” solution \hat{x} has a cost function $f(\hat{x}) > f_{th}$, where f_{th} is a pre-computed threshold value.

In the following, we give a pseudo-code description of the AntMIMO algorithm where the parameter \mathcal{T} denotes the pheromone array storing the pheromone values τ_{ij} and x^{ib} denotes the best solution returned by the ants in a single construction iteration.

Algorithm 5.2: AntMIMO

input: a problem instance P_{MIMO}
 $x^{bs} \leftarrow \text{NULL}$.
for all τ_{ij} **do**
 $\tau_{ij} \leftarrow \tau_0$
end for
while termination conditions not met **do**
 for $k \leftarrow 1$ **to** m **do**
 empty ant k 's list of visited nodes.
 $(x^k, w^k) \leftarrow \text{ConstructAntsSolutions}$
 $(x_{LS}^k, w_{LS}^k) \leftarrow \text{ApplyLocalSearch}(x^k, w^k)$
end for
 $x^{ib} \leftarrow \arg \min (f(x_{LS}^1), \dots, f(x_{LS}^m))$
if $x^{bs} = \text{NULL}$ **or** $f(x^{ib}) < f(x^{bs})$ **then** $x^{bs} \leftarrow x^{ib}$
 $\text{UpdatePheromones}(x^{bs}, w^{bs}, \mathcal{T})$
 if $f(x^{bs}) > f_{th}$ **then**
 $\text{ReinitializePheromones}(\mathcal{T})$
 $x^{bs} \leftarrow \text{NULL}$
end if
end while
output : x^{bs}

1. Algorithm Parameters

Similar to other ACO algorithms, AntMIMO's implementation involves a large set of parameters, the fine tuning of which greatly affects the performance of the algorithm. The key is to find a suitable combination of parameter values that can strike a balance between excessive *exploration* of the search space and intense *exploitation* of learned search experience, which in the worst case may lead to stagnation and producing suboptimum solution. In the following, we describe the different parameters and the values assigned to them for the implementation of AntMIMO.

1. Restart Threshold f_{th} : We exploit the statistical properties of the problem to select a suitable value for f_{th} . Recall that the cost function $f(x)$ to be minimized is equal to the Euclidean distance $\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2$. For the original transmit vector \mathbf{x}^* ,

$$f(x^*) = \|\mathbf{y} - \mathbf{H}\mathbf{x}^*\|^2 = \|\mathbf{n}\|^2 = \sum_{i=1}^{2n} \mathbf{n}_i^2 \quad (5.6)$$

where $\mathbf{n} \in \mathbb{R}^{2n}$ is AWGN and $n_i \sim N(0, \sigma^2)$. From (5.6), $f(x^*)$ is a random variable, z , having a *chi-square* χ^2 distribution with $2n$ degrees of freedom with a cumulative distribution function, $F_Z(z)$, given by

$$F_Z(z) = P(Z \leq z) = 1 - e^{-z/2\sigma^2} \sum_{k=0}^{n-1} \frac{1}{k!} \left(\frac{z}{2\sigma^2} \right)^k, \quad z \geq 0 \quad (5.7)$$

From (5.7), we can determine the range of values the optimum cost function $f(x^*)$ can take for any problem instance P_{MIMO} ; consequently, we set the restart threshold f_{th} such that $P(z > f_{th}) = \epsilon$ where ϵ is a very small positive number. For a 4×4 MIMO system, for example, the probability of the optimum cost function $f(x^*)$ taking a value greater than $3.5\sigma^2$ is extremely small, $P(f(x) \geq 3.5\sigma^2) = 0.000474$, so we can set $f_{th} = 3.5\sigma^2$. If AntMIMO returns a solution \hat{x} with cost function $f(\hat{x}) > f_{th}$, we know that the algorithm has returned an unacceptable solution and

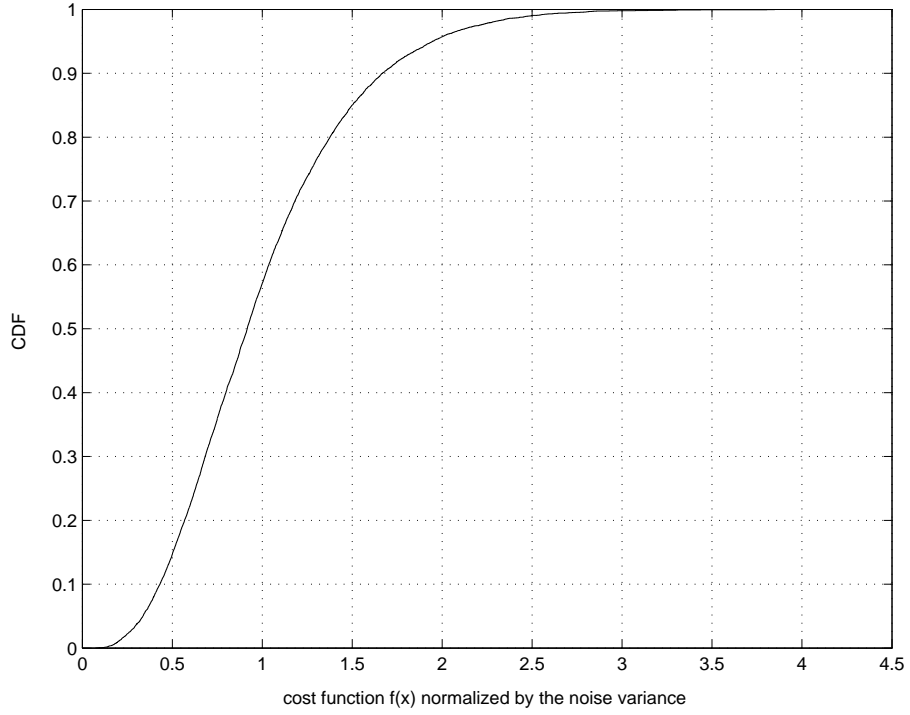


Fig. 19. CDF plot of the cost function $f(x)$ normalized by σ^2 for a 4×4 MIMO system.

needs to be restarted, Fig. 19.

2. Initial Pheromone Value τ_0 : Different ACO algorithms have different means of initializing the pheromone array. Typically, τ_0 is set to be slightly higher than the expected amount of pheromones deposited by the ants after a single iteration. The reason for this choice is that if the initial pheromone values τ_0 's are too low, the search becomes quickly biased by the first tours generated by the ants, which in general leads towards the exploration of inferior zones of the search space. On the other hand, if the τ_0 values are too high, then many iterations are lost waiting until pheromone evaporation reduces the pheromone values enough so that pheromones added by the ants can start to bias the search experience. By setting $\tau_0 = 1/f_{th}$ in AntMIMO, not only do we satisfy the general rule of thumb, but we also penalize

“bad” ants’ solutions during run time. For example, suppose we have a 4×4 MIMO system with $\sigma^2 = 1$, $\rho = 0.1$, and $f_{th} = 3.5$; by setting $\tau_0 = 1/f_{th}$, the pheromone trail values of a solution \hat{x} whose cost function is greater than f_{th} are reduced, which reduces the probability of selecting the path that led to \hat{x} in the next iteration of the algorithm. Thus, if at iteration $t = 1$, $f(x^{bs}) = 8$, then $\Delta\tau^{bs} = 0.2$ and the pheromone values become,

$$\begin{aligned}\tau_{ij}(1) &= (1 - \rho)\tau_0 + \rho\Delta\tau^{bs} && \forall e_{ij} \text{ traversed while building } x^{bs} \\ &= 0.9 * 0.285 + 0.1 * 0.125 = 0.269 < \tau_0.\end{aligned}$$

3. **Evaporation Rate ρ :** The evaporation rate $\rho \in [0,1]$ plays an important role in AntMIMO; an appropriate value for ρ is necessary to balance the *exploration* and *exploitation* phases of the algorithm. A too small value of ρ will prolong the exploration phase since in the first iterations of the algorithm, the difference in pheromone values on different edges is small and the biasing effect of the pheromone array is minimal. To exploit this biasing effect, the algorithm must run for a large number of iterations. If we select a relatively large value for ρ , the biasing effect of the pheromones becomes more pronounced in the early stages of the algorithm; we may allow shorter run time, but we run the risk that the algorithm may get stuck in not so good regions of the search space and consequently produce a suboptimum solution. In our simulations, we set $\rho \in [0.1, 0.2]$ depending on the size of the problem, a small problem size corresponds to a large value of ρ .
4. **Number of Ants m :** In the presence of local search, there is no need for a large colony of ants. In our simulations, for $n \times n$ MIMO system, we set $m = n$; increasing m beyond that doesn’t noticeably improve the performance of the algorithm.
5. **Number of iterations t :** Allowing the algorithm to run for a long period of time increases the probability of finding the optimum solution, yet we want t to be as

small as possible so that we can detect the transmitted symbols as fast as possible in a wireless transmission scenario; (5.8) gives a rough guide to choosing a suitable t .

$$t^c = \frac{\ln\left(\frac{0.01\Delta\tau^{bs}}{\Delta\tau^{bs} - \tau_0}\right)}{\ln(1 - \rho)} \quad (5.8)$$

where t^c is the time needed for the pheromone values τ_{ij}^{bs} along the path corresponding to the best-so-far solution x^{bs} to reach 99% of $\Delta\tau^{bs}$, assuming that the algorithm returns the same x^{bs} after every iteration. For a 4×4 MIMO system using $\tau_0 = 1/3.5\sigma^2$, $\rho = 0.15$, and replacing $\Delta\tau^{bs}$ by the empirical average of $\Delta\tau^* = 1.33/\sigma^2$, the time needed for the algorithm to converge to the optimum solution is $t^c \approx 27$ iterations; (5.8) can be easily derived from the global pheromone update rule of (5.4) by rewriting the latter as

$$\tau_{ij}(t) = (1 - \rho)^t \tau_0 + \Delta\tau^{bs}[1 - (1 - \rho)^t] \quad (5.9)$$

and setting $\Delta\tau_{ij}(t^c) = 0.99\Delta\tau^{bs}$. The parameter t^c gives an indication of the time needed for the algorithm to reach *stagnation*, a situation where the algorithm stops exploring new paths and keeps on choosing a single path every time. In our simulations, we set $t \leq t^c$. Finally, the number of times the algorithm is allowed to restart after producing a “bad” solution is determined by a restart counter ct . In our simulations, we set $ct = 2$.

2. Algorithm Performance

We tested the algorithm on several $n \times n$ MIMO systems in a Rayleigh fading environment as specified in Section 1.2 with transmit symbols $\tilde{\mathbf{x}}$ selected from a 4-QAM constellation.

In our simulations, the average energy per information bit² is fixed to $E_b = 1$. The transfer matrix $\tilde{\mathbf{H}}$ is modeled by independent Gaussian random variables of variance 0.5 per dimension. The variance σ^2 of the AWGN $\tilde{\mathbf{n}}$ per dimension is adjusted by the formula

$$\sigma^2 = \frac{nE_{s_{av}}}{2\log_2(q)} 10^{\frac{-SNR_b}{10}} \quad (5.10)$$

where $E_{s_{av}}$ is the average symbol energy of the M -QAM constellation when $E_b = 1$ and SNR_b is the signal-to-noise ratio per information bit.

a. Simulation Results

Fig. 20 compares the performance of the AntMIMO algorithm to the optimum Maximum-Likelihood performance for 2×2 , 4×4 , and 6×6 MIMO systems employing 4-QAM constellations. We notice that AntMIMO, in the three cases, achieves a near ML performance; the BER plots of AntMIMO are only 0.5 dB worse than those corresponding to the ML detector; note that AntMIMO can almost approach ML performance, if we allow it to run for a long period of time.

The different parameter sets used in the simulations are specified in Table. I.

Table I. Parameter Settings Used in Simulations

MIMO System	m	t	ρ	f_{th}	ct
2×2	2	4	0.2	$2\sigma^2$	2
4×4	4	10	0.15	$3\sigma^2$	2
6×6	6	16	0.1	$3.5\sigma^2$	2

Fig. 21 illustrates the bit error rate performance comparison of the AntMIMO de-

²This normalization lets the elements of x to be selected from the set $\mathbb{Z}_{\sqrt{M}}$, which is useful when implementing the search tree.

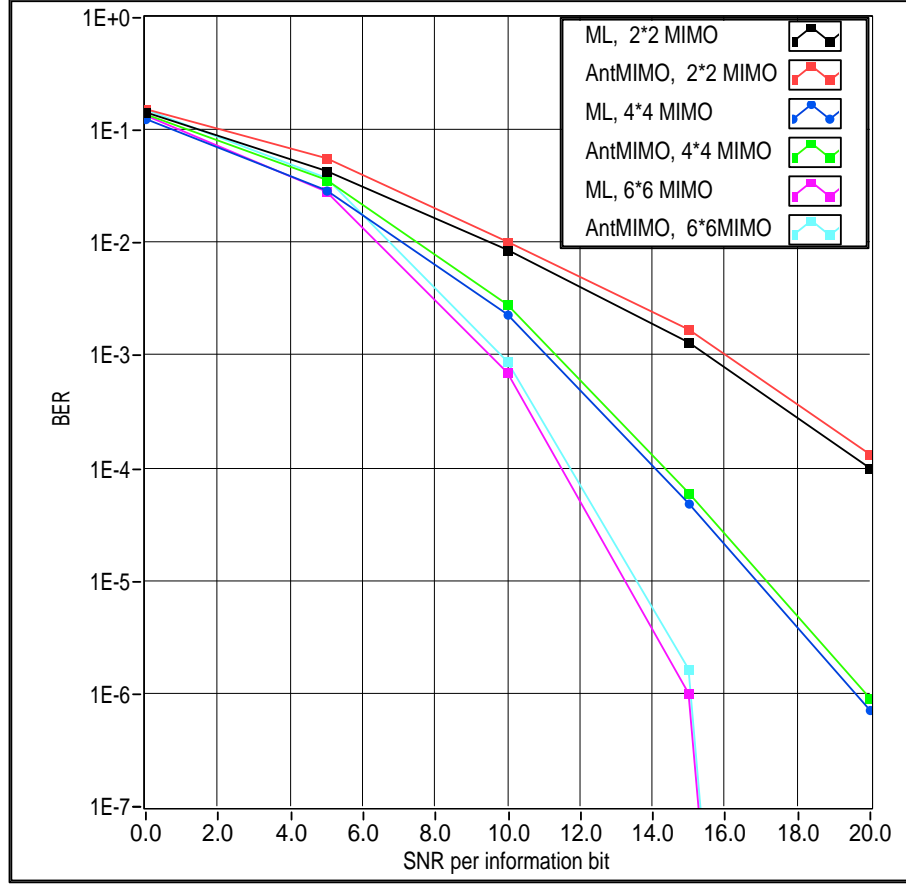


Fig. 20. AntMIMO vs. Maximum-Likelihood BER plots.

coder, the sphere decoder, the ZF-decoder, and the MMSE-decoder for a 4×4 MIMO system employing 4-QAM. Clearly, AntMIMO outperforms the linear decoders and is slightly worse than the sphere decoder's performance, which is maximum-likelihood.

b. Effect of Parameter Settings on AntMIMO's Performance

In this section, we investigate the influence of the various parameter settings and action choice rules on AntMIMO's performance. We first study the effect of the ant colony's size on AntMIMO's performance. Fig. 22 shows the BER plots of a 4×4 MIMO system,

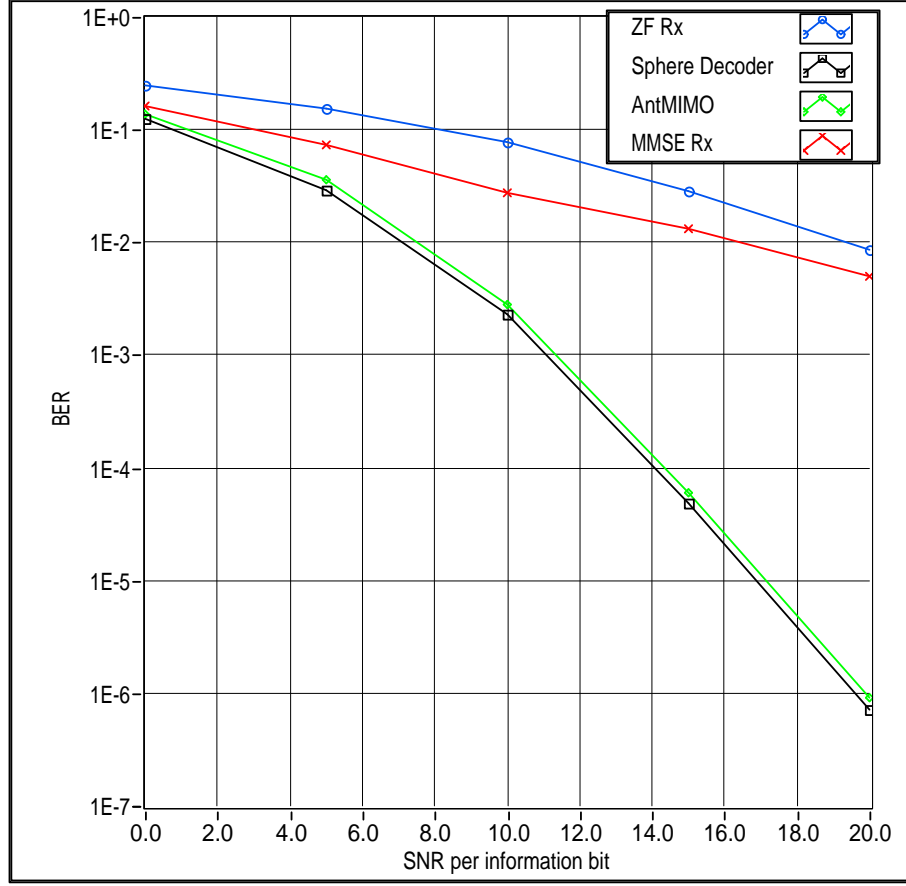


Fig. 21. AntMIMO vs. SD, ZF, and MMSE decoders for a 4×4 MIMO system with 4-QAM.

using 4-QAM with $\rho = 0.15$, $t = 12$, $ct = 2$, $f_{th} = 3\sigma^2$ and $m \in \{1, 2, 4, 8\}$. As expected, AntMIMO with a single ant has the worst performance; the performance is improved when using two ants instead of one, but the best compromise between complexity and performance is attained when m is selected to be in the order of n for an $n \times n$ MIMO system. Using a larger colony of ants doesn't improve the performance much as apparent in the case when $m = 8$.

Fig. 23 illustrates the effect of the evaporation rate on the performance of a 4×4 MIMO system, employing 4-QAM constellation with $m = 4$, $t = 10$, $ct = 2$, $f_{th} = 3\sigma^2$ and $\rho \in \{0.01, 0.15, 0.5\}$. Using a run time of 10 iterations with the small evaporation

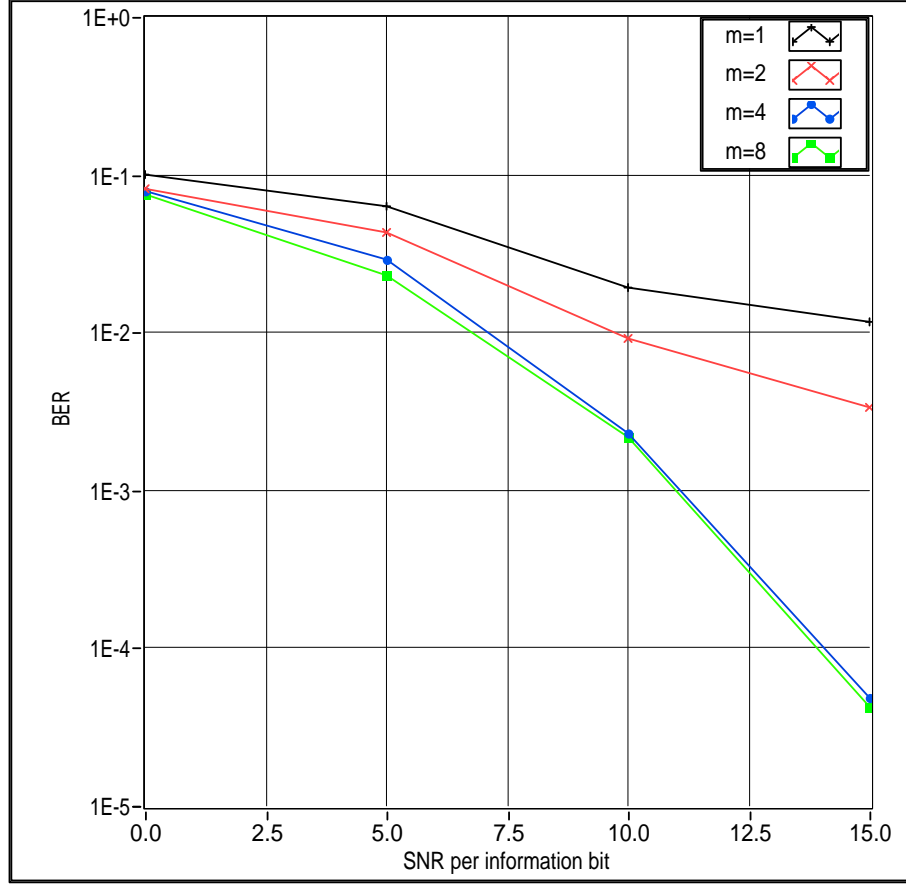


Fig. 22. Influence of the ant colony's size on the performance of AntMIMO for a 4×4 MIMO system with 4-QAM constellation.

rate of $\rho = 0.01$ results in small differences in the pheromone values with time; hence the positive feedback effect of the pheromones is not evident, and this combination has the worst performance among the three BER curves. Likewise, having a too high evaporation factor of $\rho = 0.5$ leads to the rapid convergence of the algorithm to suboptimum solutions leading to an inferior performance to the case when $\rho = 0.15$.

Recall that in the global pheromone update rule of (5.4), we allow only the best-so-far found solution x^{bs} to update its pheromone trails. In the literature, some of the algorithms such as the *MMAS* algorithm allow the ant that returned the best solution at the end of a

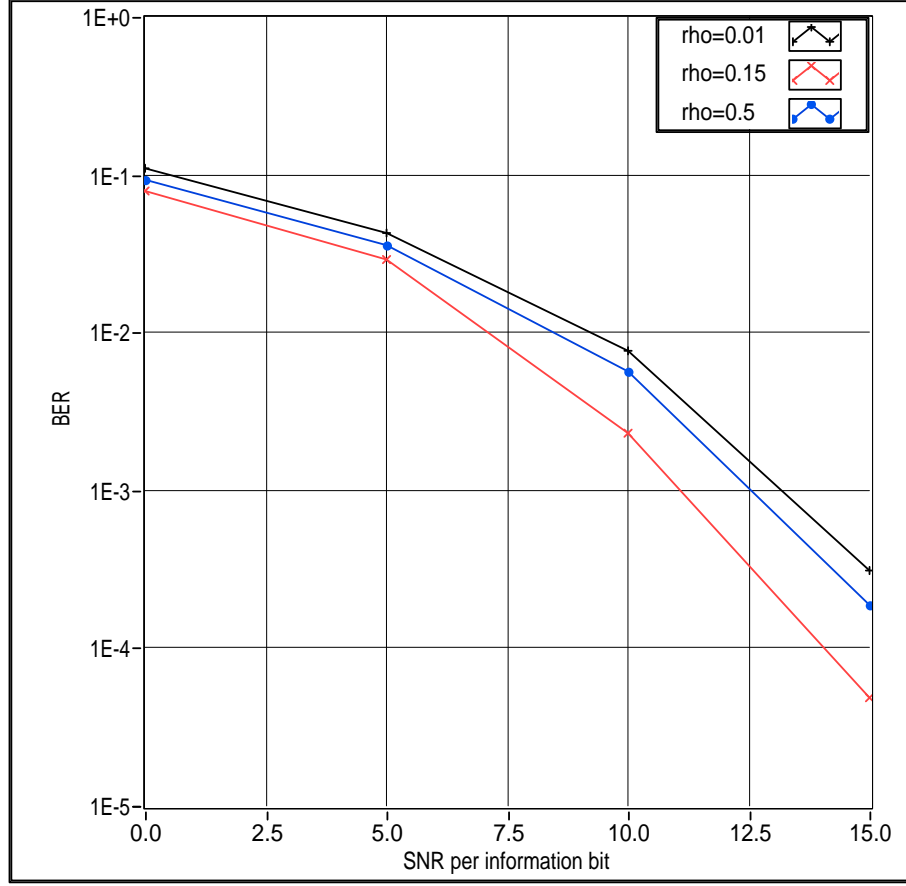


Fig. 23. Influence of the evaporation rate on the performance of AntMIMO for a 4×4 MIMO system with 4-QAM constellation.

construction iteration x^{ib} to update its pheromone trails instead. For the MIMO detection problem, updating the pheromone trails of x^{bs} instead of x^{ib} yields a better performance, as illustrated in Fig. 24.

We also noticed that better performance is achieved when incorporating a restart mechanism as illustrated in Fig. 25, and when using the 1-opt local search algorithm. As evident in Fig. 26, AntMIMO without a local search algorithm performs significantly worse than when a local search algorithm is incorporated given that the same set of parameters is used in both scenarios, $m = 4$, $t = 12$, $\rho = 0.15$, $ct = 2$. For the version without local

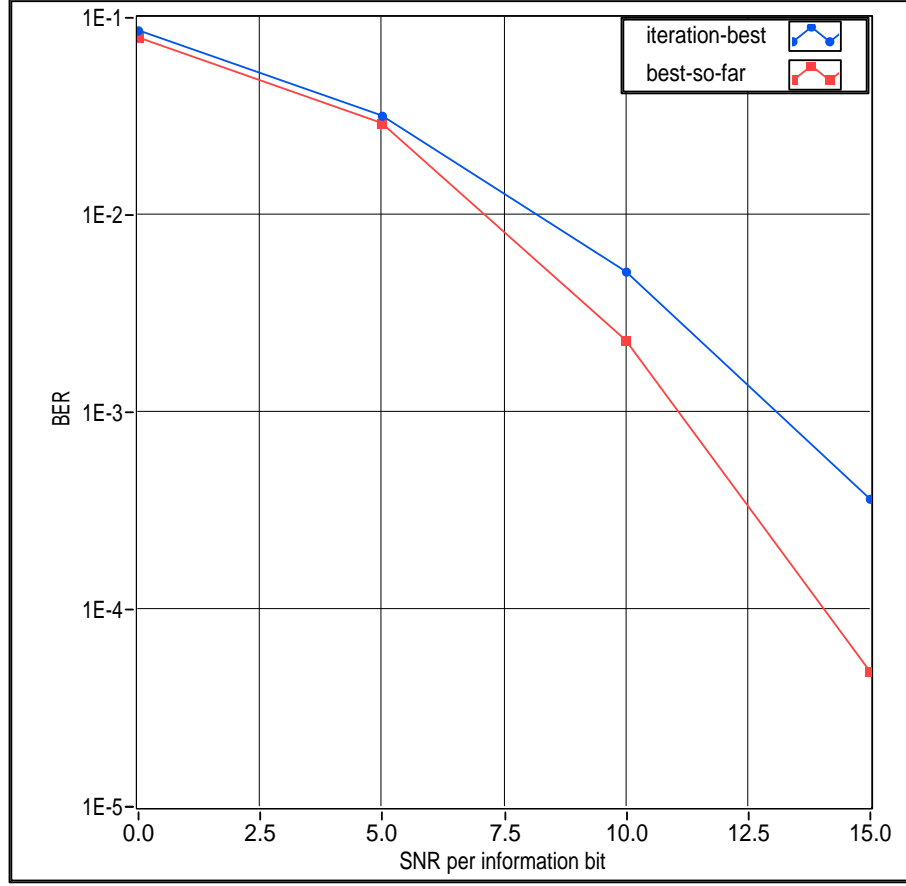


Fig. 24. Performance of iteration-best update rule vs. best-so-far update rule for a 4×4 MIMO system using 4-QAM constellation.

search to achieve the same performance as that with a local search, we had to adjust the evaporation rate and increase significantly both the size of the colony and the number of iterations, $\rho = 0.1$, $m = 10$, and $t = 45$.

In Chapter IV, we mentioned that the dynamic change of the pheromone array values at run-time is extremely important. It serves as a guiding mechanism that manipulates the lower level constructive and local search heuristics into concentrating their search on “good” regions of the search space. Figs. 27, 28, 29 demonstrate this dynamic change of pheromone values during run-time and how they influence solution

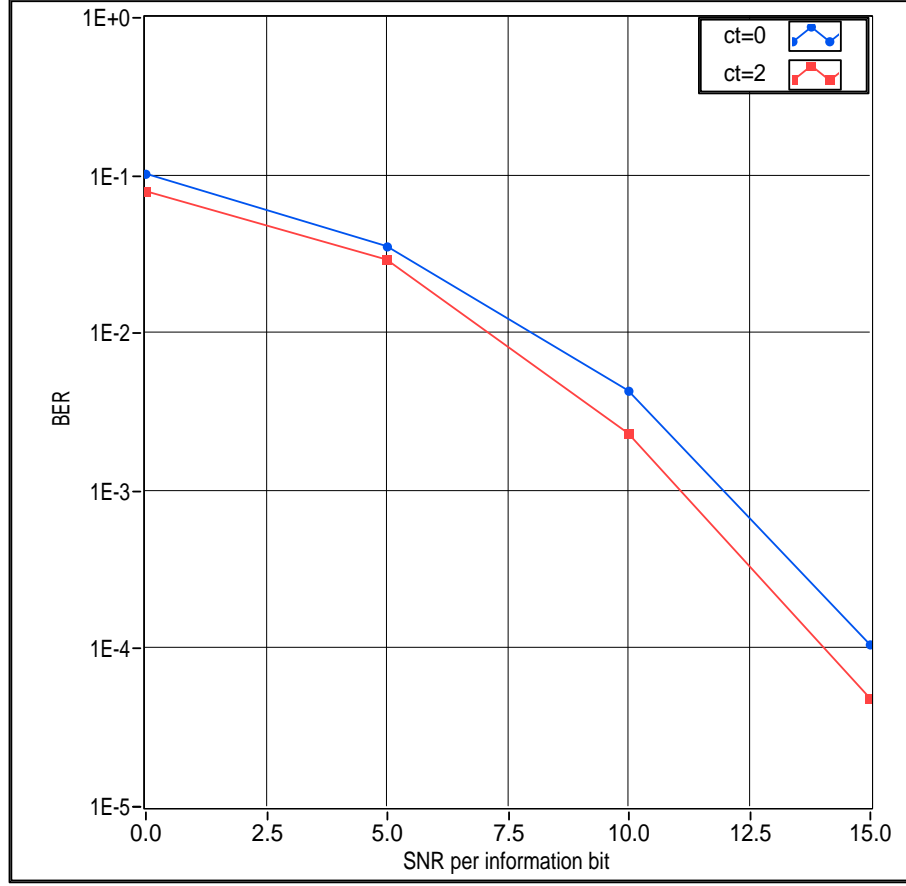


Fig. 25. Performance of a restart mechanism with $ct = 2$ vs. no restart mechanism for a 4×4 MIMO system using 4-QAM constellation.

construction for a 4×4 MIMO system using 4-QAM constellation, $\rho = 0.15$, $m = 4$, $f_{th} = 3\sigma^2$, $ct = 2$, and $t \in \{4, 12, 25\}$. For this instance of P_{MIMO} , the optimum solution x^* has a cost function $f(x^*) = 0.199425$ and a corresponding path $w^* = \langle c_{2,1}, c_{5,0}, c_{11,0}, c_{23,0}, c_{47,0}, c_{95,0}, c_{162,1}, c_{385,0} \rangle$; so we expect as $t \rightarrow \infty$, the pheromone values, τ_{ij}^* along the path leading to the optimum solution to reach $\Delta\tau^* = 1/f(x)^* = 1.00289$. We observe from the stem plots, that the pheromone values corresponding to a good solution are re-enforced with time. After 4 iterations, the difference in pheromone values on different branches is not that big, but after 12 iterations the difference is clearly apparent,

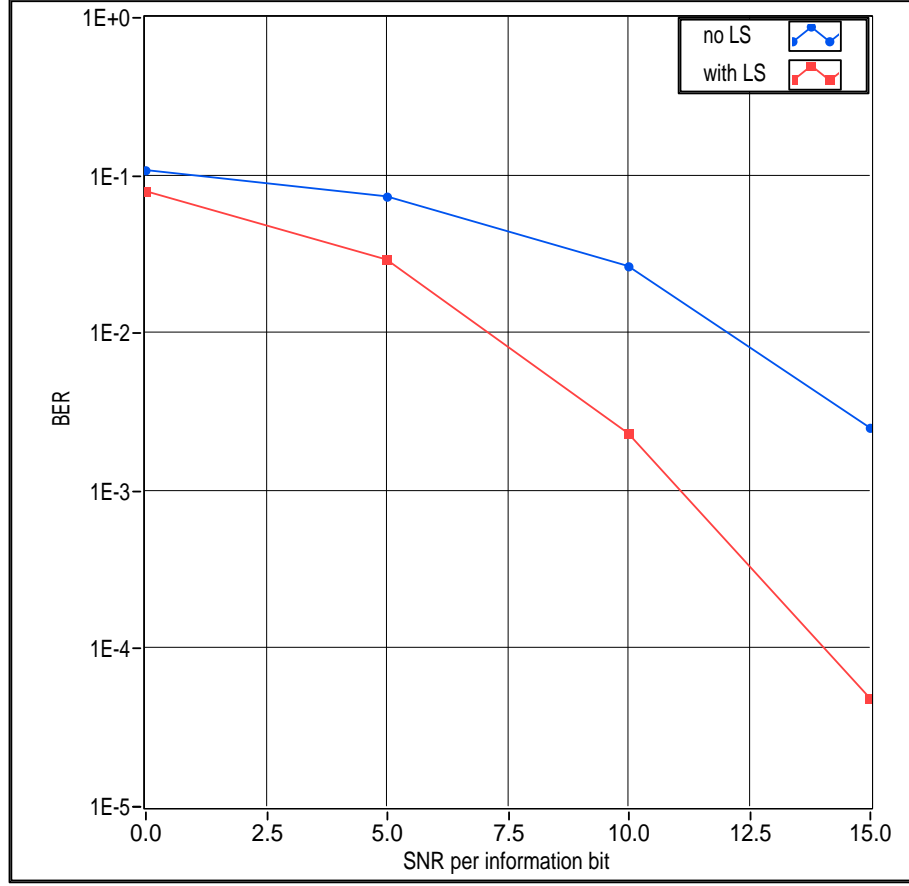


Fig. 26. Performance of AntMIMO plus local search vs. AntMIMO minus local search for a 4×4 MIMO system using 4-QAM constellation.

which indicates that with very high probability, the ants will select the path with the higher amount of pheromones. After 25 iterations, the pheromone values on the path corresponding to the best-so-far solution have almost reached $\Delta\tau_{ij}^*$, $\Delta\tau_{ij}^{bs} = 0.988551$. Note also that, in all three cases, bad solutions with cost function greater than f_{th} are penalized during run time by decreasing the pheromone values on their corresponding edges.

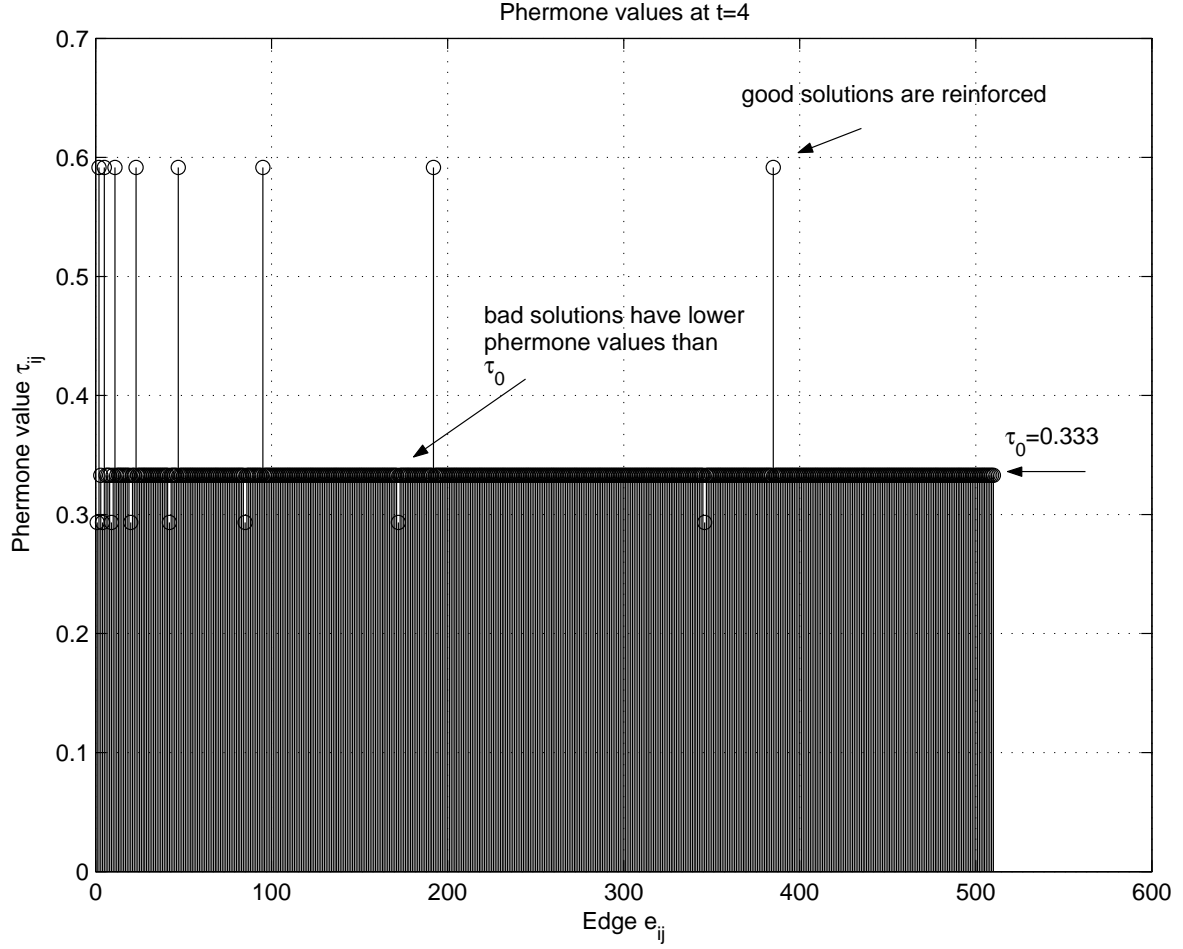


Fig. 27. Pheromone array stem plot at $t = 4$ for a 4×4 MIMO system with 4-QAM constellation.

c. Complexity of the AntMIMO Algorithm

MIMO detection algorithms, such as the sphere decoder or the nulling and cancelling decoder, typically require some form of preprocessing, such as QR factorization or lattice reduction, to be performed on the channel matrix $\tilde{\mathbf{H}}$ before applying the detection algorithm. The computational complexity of such procedures is often discounted from the computational complexity of these algorithms. In AntMIMO, there is no need for any such preprocessing stage; the computational complexity of the algorithm is dominated by

the complexity of the local search heuristic used. A naive implementation of the *1-opt* local search heuristic used in AntMIMO is of cubic complexity; in our implementation, instead of computing the cost of each new candidate solution x' , we only calculate the differential gain that can be achieved when choosing x' instead of x^k ; the complexity of this implementation is quadratic. The resulting computational complexity of AntMIMO is also quadratic, $O(mtn^2)$ for an $n \times n$ MIMO system using 4-QAM constellation, m ants, and t iterations.

d. Convergence of the AntMIMO Algorithm

Currently, we do not have a theoretical result on the convergence of the AntMIMO algorithm. Yet, experimentally, the algorithm exhibits both convergence in value and convergence in solution. We noticed that if we allow the algorithm to run for a reasonable number of iterations, it produces the optimum solution at least once, hence the convergence in value. On the other hand, if we set $\rho = 0.01$ and let the algorithm run for a very long period of time, the algorithm shows convergence in solution; all the ants select the path leading to the optimum solution at the end of each iteration t , $t > t^c$.

e. Fitness Landscape Analysis of Problem P_{MIMO}

The theory of fitness landscapes [52, 53], originally developed to provide a mathematical framework for studying the dynamics of biological evolutionary optimization, has been shown to be very useful for understanding the behavior and performance of combinatorial optimization algorithms. In this section, we perform a fitness landscape analysis on problem P_{MIMO} in order to study the effectiveness of the AntMIMO algorithm when applied to P_{MIMO} .

Simply speaking, a fitness landscape is a representation of the search space; it can be visualized as a mountainous region with hills, craters, and valleys. In a minimization

problem, a heuristic algorithm can be thought of as navigating through this landscape to find its lowest point. More formally, a fitness landscape (S, f, d) of a combinatorial optimization problem instance consists of the set of all feasible solutions S , an objective function $f : S \rightarrow \mathbb{R}$, which assigns a fitness value $f(x)$ to every $x \in S$, and a distance measure d which defines the spatial structure of the landscape; the distance $d(x, x')$ between two solutions x and x' can be defined as the minimum number of moves that have to be performed to transform x into x' . When analyzing $n \times n$ MIMO systems employing 4-QAM constellation, we define the fitness function $f(x)$ to be equal to the cost function $\|y - Hx\|^2$ and the distance $d(x, x')$ to be the hamming distance between binary vectors x and x' .

It was found that the distribution of local minima in the landscape and their relative location with respect to global optima is an important criterion for studying the effectiveness of adaptive multi-start algorithms like ACO algorithms [45]. The Fitness Distance Correlation (FDC) is an important measure that determines how closely solution fitness and the distance to the global optima are related, [52]

$$\varrho(F, D) = \frac{\text{Cov}(F, D)}{\sigma(F)\sigma(D)} \quad (5.11)$$

where $\text{Cov}(F, D)$ is the covariance between the random variables F and D which probabilistically describe the fitness and the distance of local optima to a global optimum, while $\sigma(F)$ and $\sigma(D)$ denote their standard deviations. For minimization problems, a high positive correlation factor $\varrho(F, D)$ indicates that the smaller the solution cost becomes, the closer are the solutions - on average - to a global optimum. For such a case, the search is expected to be “easy” and algorithms combining adaptive solution generation and local search, such as ACO algorithms, are expected to perform well. In AntMIMO, for instance, the smaller the cost of the found solution x^{bs} , the higher the amount of pheromones that will be deposited along its path; this means that in the next iteration of the algorithm,

the search will concentrate in the vicinity of x^{bs} . If problem instance P_{MIMO} has a high positive fitness distance correlation factor, then the global optimum x^* is close to x^{bs} , and there is a high probability that AntMIMO will output x^* . Fig. 30 shows the fitness distance scatter plot of a P_{MIMO} instance with an FDC factor $\varrho = 0.941$. We notice that the solutions with hamming distance $d = 1$, which are closest to the global minimum, are also the ones with the lower cost functions. The further we get from the global optimum, the larger the solution cost becomes.

We empirically computed the FDC factor for 2×2 , 4×4 , and 6×6 MIMO systems using a 4-QAM constellation at different values of SNR. In all instances, problem P_{MIMO} exhibited a relatively high positive mean correlation factor, $\varrho \in [0.66, 0.73]$; we also noticed that ϱ increases with increasing SNR. From this analysis, we expect AntMIMO to perform well on problem P_{MIMO} , and this is actually the case as evident from earlier simulations.

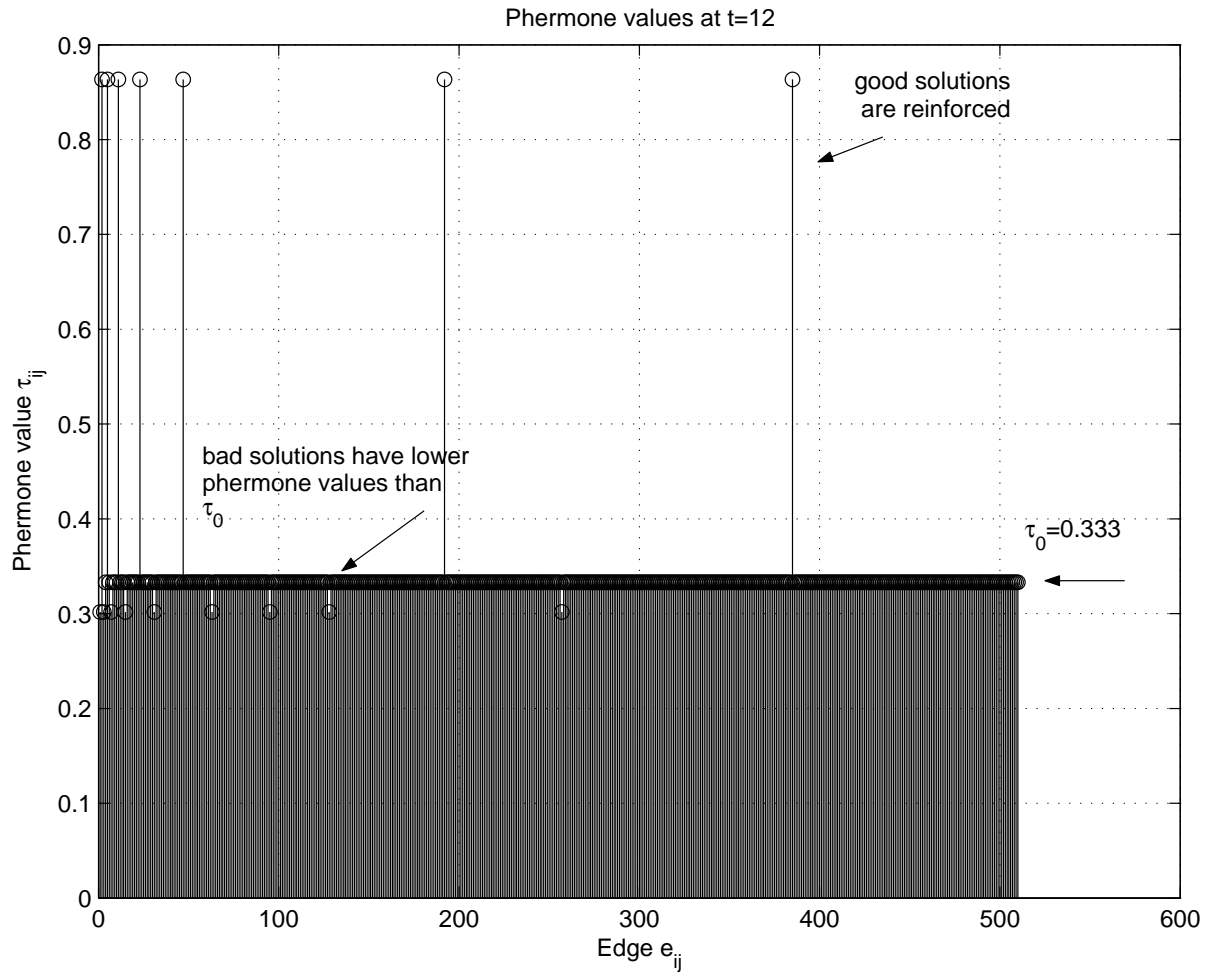


Fig. 28. Phermone array stem plot at $t = 12$ for a 4×4 MIMO system with 4-QAM constellation.

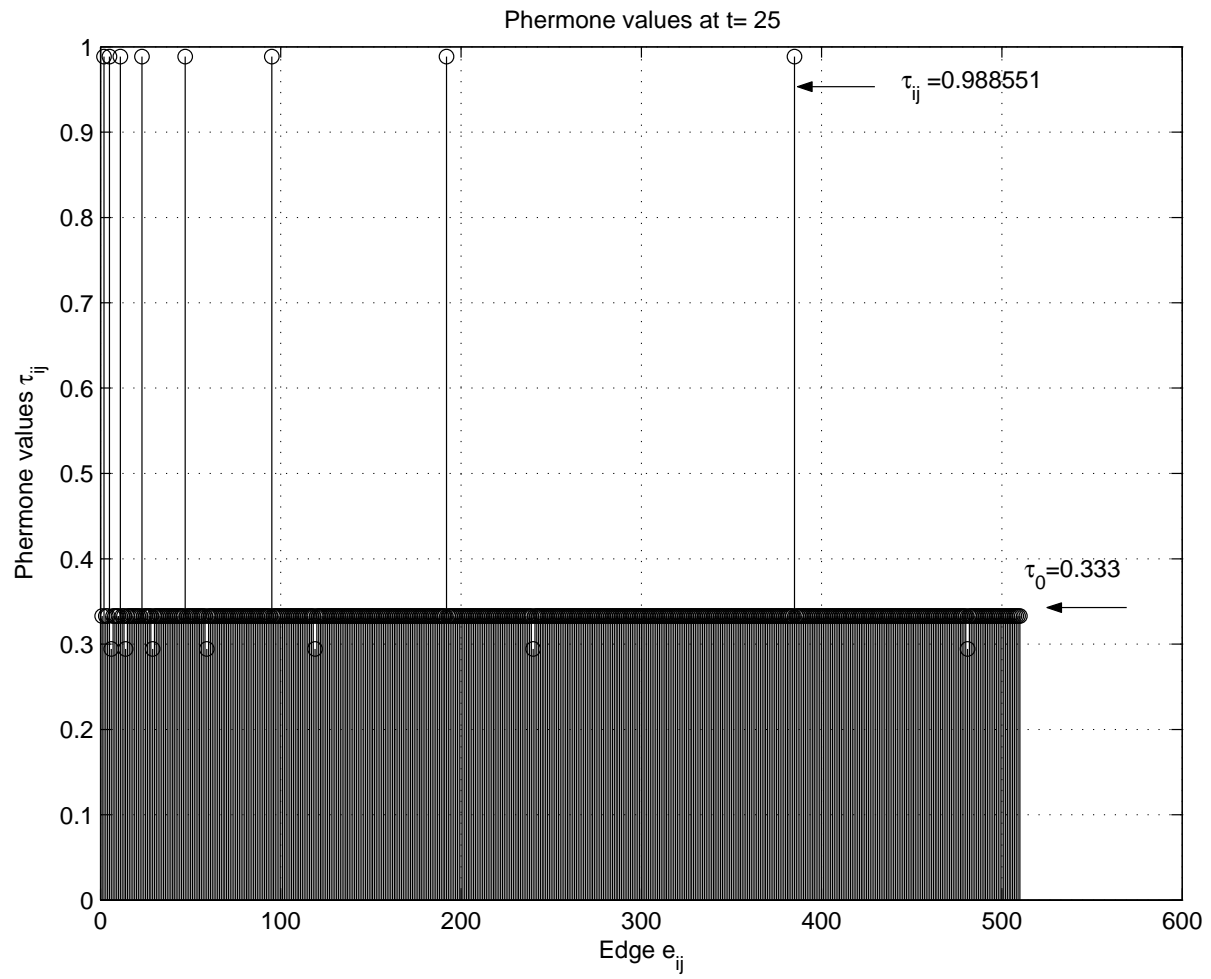


Fig. 29. Pheromone array stem plot at $t = 25$ for a 4×4 MIMO system with 4-QAM constellation.

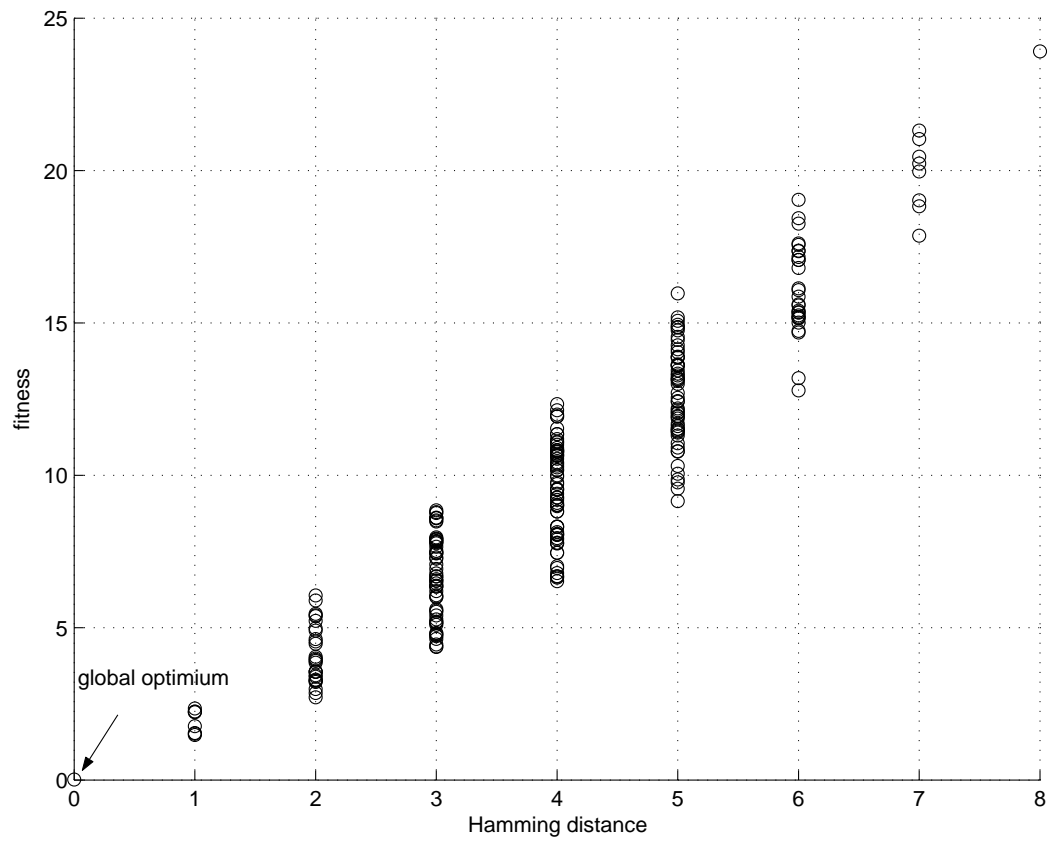


Fig. 30. Fitness distance scatter plot with $\varrho = 0.941$.

CHAPTER VI

SUMMARY

In this thesis, we considered the MIMO receiver design problem in a spatial multiplexing scheme. The MIMO detection problem is equivalent to solving the integer least-squares problem which is *NP*-hard. In the literature, suboptimal detection algorithms, typically of polynomial complexity, are often employed, while exact algorithms, like the sphere decoder algorithm, that solve the MIMO detection problem to optimality have an average exponential complexity. The main contribution of this thesis is the design of an approximate algorithm, AntMIMO, that outperforms other heuristic techniques and still exhibit polynomial complexity when used for the detection of MIMO systems. AntMIMO belongs to the family of Ant Colony Optimization metaheuristic, which is a recent algorithmic technique that has been inspired by the behavior of real ants to develop strategies for solving hard combinatorial optimization problems. A formal introduction to the ACO paradigm was provided in Chapter IV. In Chapter V, we provided a detailed description of the AntMIMO algorithm. We showed that it is indeed an effective technique for MIMO detection by performing fitness landscape analysis on the integer least-squares problem and verifying the existence of strong positive correlation between solution fitness and distance to the global optimum. The effectiveness of AntMIMO was further demonstrated by simulating its performance on several $n \times n$ MIMO systems using a 4-QAM constellation; AntMIMO's BER curves were only 0.5 dB worse than the ML curves. We also studied the influence of various parameter settings on AntMIMO's performance and estimated its computational complexity to be quadratic in the number of used antennae.

A. Future Work

The AntMIMO algorithm has been optimized for the detection of MIMO systems using a 4-QAM constellation. Typical MIMO configurations involve the use of higher order constellations such as 16-QAM. The complexity of AntMIMO is expected to increase as a more sophisticated local search algorithm will be incorporated within AntMIMO.

Another interesting problem is to configure AntMIMO for the detection of coded MIMO transmissions. Coding constraints can be implemented by the ants' construction policy on the associated graph.

Finally, in this thesis, we equated the MIMO detection problem to a combinatorial optimization problem that is readily solvable by using a heuristic technique. Another interesting perspective would be to solve the MIMO detection problem through the use of convex optimization techniques. We briefly investigated the use of the Reformulation-Linearization Technique (RLT) of Sherali and Adams [54] for MIMO detection. The integer least-squares problem was reformulated as a binary mixed integer program that was solved using the CPLEX solver. Although the results that we obtained were not as competitive as those of AntMIMO's, we believe that such a technique warrants further investigation.

REFERENCES

- [1] W. C. Jakes, *Microwave Mobile Communications*, New York: John Wiley and Sons, 1974.
- [2] T. Marzetta, “BLAST training: estimating channel characteristics for high-capacity space-time wireless,” in *Proceedings of Allerton Conference on Communication, Control and Computing*, Monticello, IL, September 1999, pp. 22–24.
- [3] B. Hassibi and B. Hochwald, “Optimal training in space-time systems,” in *Proceedings of Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, October 2000, pp. 743–747.
- [4] E. Telatar, “Capacity of multi-antenna in Gaussian channels,” *European Transactions on Telecommunications*, vol. 10, pp. 585–595, November 1999.
- [5] G. J. Foschini and M. J. Gans, “On the limits of wireless communications in a fading environment when using multiple antennae,” *Wireless Personal Communications*, vol. 3, pp. 311–355, March 1998.
- [6] G. J. Foschini, “Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas,” *Bell Labs Technical Journal*, vol. 1, pp. 41–59, 1996.
- [7] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, New York: John Wiley and Sons, 1991.
- [8] A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications*, New York: Cambridge University Press, 2003.

- [9] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, “V-BLAST: an architecture for realizing very high data rates over the rich-scattering wireless channel,” in *International Symposium on Signals Systems and Electronics*, Pisa, Italy, September 1998, pp. 295–300.
- [10] S. M. Alamouti, “Simple transmit diversity technique for wireless communications,” *IEEE Journal on Select Areas in Communications*, vol. 16, pp. 1451–1458, 1998.
- [11] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, “Space-time block codes for orthogonal designs,” *IEEE Transactions on Information Theory*, vol. 45, pp. 1456–1466, July 1996.
- [12] V. Tarokh, N. Seshadri, and A. R. Calderbank, “Spacetime codes for high data rate wireless communication: Performance analysis and code construction,” *IEEE Transactions on Information Theory*, vol. 44, pp. 744–765, March 1998.
- [13] S. Verdu, *Multuser Detection*, New York: Cambridge University Press, 1998.
- [14] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd edition, Baltimore: John Hopkins University Press, 1996.
- [15] M. Grotschel, L. Lovasz, and A. Schriver, *Geometric Algorithms and Combinatorial Optimization*, 2nd edition, Baltimore: Springer Verlag, 1993.
- [16] S. Verdu, “Computational complexity of optimum multiuser detection,” *Algorithmica*, vol. 4, pp. 303–312, 1989.
- [17] M. Dorigo and T. Stutzle, *Ant Colony Optimization*, Cambridge, MA: MIT Press, 2004.
- [18] U. Fincke and M. Pohst, “Improved methods for calculating vectors of short length in a lattice, including a complexity analysis,” *Mathematics of Computation*, vol. 44,

- pp. 463–471, April 1985.
- [19] E. Viterbo and J. Boutros, “A universal lattice code decoder for fading channels,” *IEEE Transactions on Information Theory*, vol. 45, pp. 1639–1642, July 2000.
 - [20] M. O. Damen, H. EL Gamal, and G. Caire, “On maximum-likelihood detection and the search for the closest lattice point,” *IEEE Transactions on Information Theory*, vol. 49, pp. 2389–2402, October 2003.
 - [21] H. Vikalo, “Sphere decoding algorithms for digital communications,” Ph.D. dissertation, Stanford University, Stanford, CA, 2003.
 - [22] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, “Closest point search in lattices,” *IEEE Transactions on Information Theory*, vol. 48, pp. 2201–2214, August 2002.
 - [23] W. Zhao and G. B. Giannakis, “Sphere decoding algorithms with improved radius search,” *IEEE Transactions on Communications*, vol. 53, pp. 1104–1109, July 2005.
 - [24] M. Stojnic, H. Vikalo, and B. Hassibi, “A branch and bound approach to speed up the sphere decoder,” in *ICASSP, Philadelphia, PA*, pp. 429–432, 2005.
 - [25] J. Jalden and B. Ottersen, “On the complexity of sphere decoding in digital communications,” *IEEE Transactions on Signal Processing*, vol. 53, pp. 1474–1484, 2005.
 - [26] Z. Xie, R. T. Short, and C. K. Rushforth, “A family of suboptimum detectors for coherent multi-user communications,” *IEEE Journal on Selected Areas in Communications*, vol. 8, pp. 683–690, May 1990.
 - [27] M. Honig, U. Madhow, and S. Verdu, “Blind adaptive multiuser detection,” *IEEE Transactions on Information Theory*, vol. 41, pp. 994–960, July 1995.
 - [28] B. Hassibi, “An efficient square-root algorithm for BLAST,” Technical report, Bell Labs, Murray Hill, NJ, 1999.

- [29] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*, Englewood Cliffs, NJ: Prentice-Hall, 2000.
- [30] A. Mobasher, M. Taherzadeh, R. Sotirov, and A. K. Khandani, "A near maximum likelihood decoding algorithm for mimo systems based on semi-definite programming," Technical report, University of Waterloo, Waterloo, Canada, July 2005.
- [31] A. Weisel, Y. C. Eldar, and S. Shamai, "Semidefinite relaxation for detection of 16-qam signaling in mimo channels," *IEEE Signal Processing Letters*, vol. 12, no. 9, pp. 653–656, September 2005.
- [32] H. Wolkowicz, R. Saigal, and L. Vandenberghe, *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*, Norwell, MA: Kluwer, 2000.
- [33] S. Boyd and L. Vandenberghe, *Introduction to Convex Optimization with Engineering Applications*, Stanford, CA: Stanford University Press, 2003.
- [34] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, New York: W.H. Freeman and Co., 1979.
- [35] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers and Operational Research*, vol. 13, pp. 533–549.
- [36] F. Glover and G. Kochenberger, *Handbook of Metaheuristics*, Norwell, MA: Kluwer, 2002.
- [37] J. Drezo, A. Petrowski, P. Siarry, and E. Taillard, *Metaheuristics for Hard Optimization: Methods and Case Studies*, Berlin: Springer-Verlag, 2006.
- [38] P. H. Tan and L. K. Rasmussen, "Multiuser detection in CDMA: A comparison of relaxations, exact, and heuristic search methods," *IEEE Transactions on Wireless Communications*, vol. 3, no. 5, pp. 1802–1809, September 2004.

- [39] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, New York: Oxford University Press, 1999.
- [40] R. K. Vander Meer, M. D. Breed, M. L. Winston, , and K. E. Espelie, *Pheromone Communication in Social Insects: Ants, Wasps, Bees, and Termites*, Boulder, CO: Westview Press, 1997.
- [41] R. Beckers, J. L. Deneubourg, and S. Goss, “Trails and u-turns in the selection of a path by the ant *lasius niger*,” *Journal of Theoretical Biology*, vol. 159, pp. 397–415, 1992.
- [42] M. Dorigo, V. Maniezzo, and A. Colorni, “Ant system: optimization by a colony of cooperating agents,” *IEEE Transactions on System, Man, and Cybernetics*, vol. 26, pp. 29–41, February 1996.
- [43] Marco Dorigo, “Ant colony optimization web page,” <http://iridia.ulb.ac.be/mdorigo/ACO/ACO.html>, March 2006.
- [44] M. Dorigo and L. M. Gambardella, “Ant colony system: A cooperative learning approach to the traveling salesman problem,” *IEEE Transactions on Evolutionary Computation*, vol. 1, pp. 53–66, 1997.
- [45] T. Stutzle and H. H. Hoos, “Max-min ant system,” *Future Generation of Computer Systems*, vol. 16, pp. 889–914, 2000.
- [46] V. Maniezzo, “Exact and approximate non-deterministic tree search procedures for the quadratic assignment problem,” *INFORMS Journal for Computing*, vol. 11, pp. 358–369, 1999.
- [47] M. Dorigo and T. Stutzle, *Handbook of Metaheuristics*, pp. 251–285, Norwell, MA: Kluwer, 2003.

- [48] M. Pilat and T. White, “Using genetic algorithms to optimize acs-tsp,” in *Third International Workshop on Ant Algorithms ANTS*. 2002, pp. 282–287, London: Springer-Verlag.
- [49] M. Randall, “Near parameter free ant colony optimisation,” in *Fourth International Workshop on Ant Algorithms ANTS 2004*. 2004, pp. 374–381, Brussels: Springer-Verlag.
- [50] W. Gutjahr, “A generalized convergence result for the graph-based ant system meta-heuristic,” *Probability in the Engineering and Informational Sciences*, vol. 17, pp. 545–569, 2003.
- [51] P. Merz and B. Freisleben, “Genetic algorithms for binary quadratic programming,” in *Conference on Genetic and Evolutionary Computation*. 1999, pp. 417–424, Orlando, FL: Morgan Kaufmann.
- [52] T. Jones and S. Forrest, “Fitness distance correlation as a measure of problem difficulty for genetic algorithms,” in *Sixth International Conference on Genetic Algorithms*. 1995, pp. 184–192, San Francisco, CA: Morgan Kaufmann.
- [53] S. Wright, “The roles of mutation, inbreeding, crossbreeding, and selection in evolution,” in *Sixth Congress on Genetics*, Ithaca, NY, 1932, pp. 356–366.
- [54] H. Sherali and W. Adams, *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*, Dordrecht, Netherlands: Kluwer Academic Publishers, 1999.

APPENDIX A

ANTMIMO VIRTUAL INSTRUMENT

This appendix documents the implementation of the AntMIMO algorithm of Chapter 5 as a LABVIEW Virtual Instrument (VI).

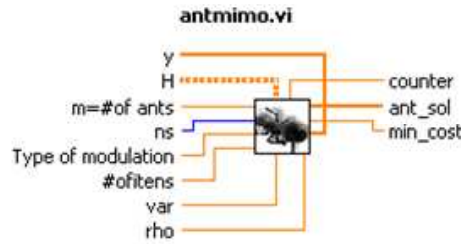


Fig. 31. Schematic representation of the AntMIMO VI.

Given a problem instance P_{MIMO} , the inputs to the AntMIMO VI are: the channel matrix H , the received vector y , the modulation type, the number of antennae, and the noise variance σ^2 , in addition to the various algorithm parameters that include the number of ants m , the number of iterations t , the evaporation rate ρ , the restart counter ct , and the restart threshold f_{th} . The VI outputs the estimated transmitted sequence \hat{x} and the associated cost $f(\hat{x})$.

Figs. [32,33] show the front panel and the block diagram of the AntMIMO VI respectively.

Following the pseudo-code of Algorithm 5.2, the AntMIMO VI makes use of five main VIs: the Initialization VI, the Construction VI, the Cost Evaluation VI, the Local Search VI, and the Pheromone Update VI.

- Initialization VI

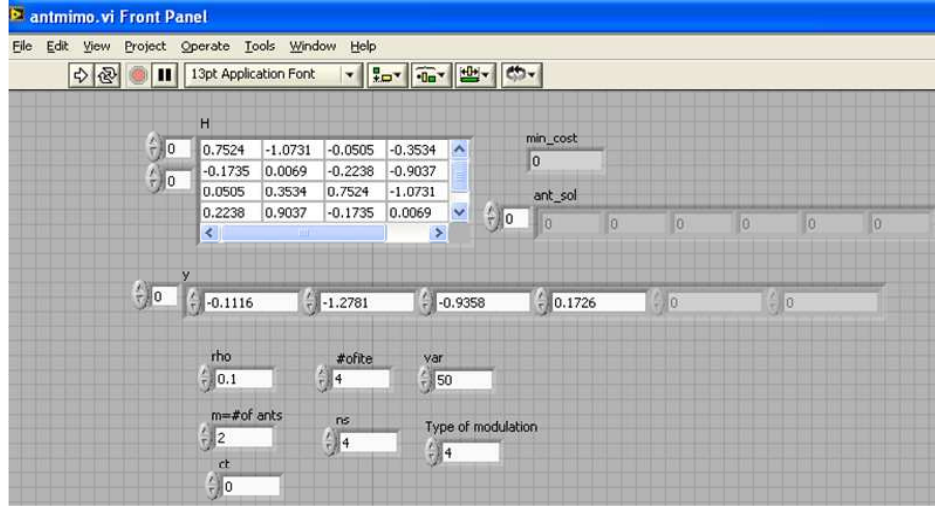


Fig. 32. Front panel of the AntMIMO VI.

This VI computes the total number of nodes present in the construction tree according to the formula,

$$N = \sum_{i=0}^{ns+1} (\sqrt{M})^i$$

where $ns = 2n$ for an $n \times n$ MIMO system employing an M -QAM constellation. It then builds a pheromone array of size N and initializes it to τ_0 . It also creates m ant structures where each structure is capable of storing the constructed solution in a $1 \times ns$ array, the associated cost function, and the list of visited nodes in a $1 \times ns$ array.

- Construction VI

This VI allows the ants to walk down the tree in parallel to construct their solutions. It first places the ants on the first level of the tree in a random fashion. Then the ants choose the next nodes to move into according to the probability choice rule of (5.3), until all ants have constructed their solutions. The probability choice rule is implemented in the subVI, ant-choose-pr,

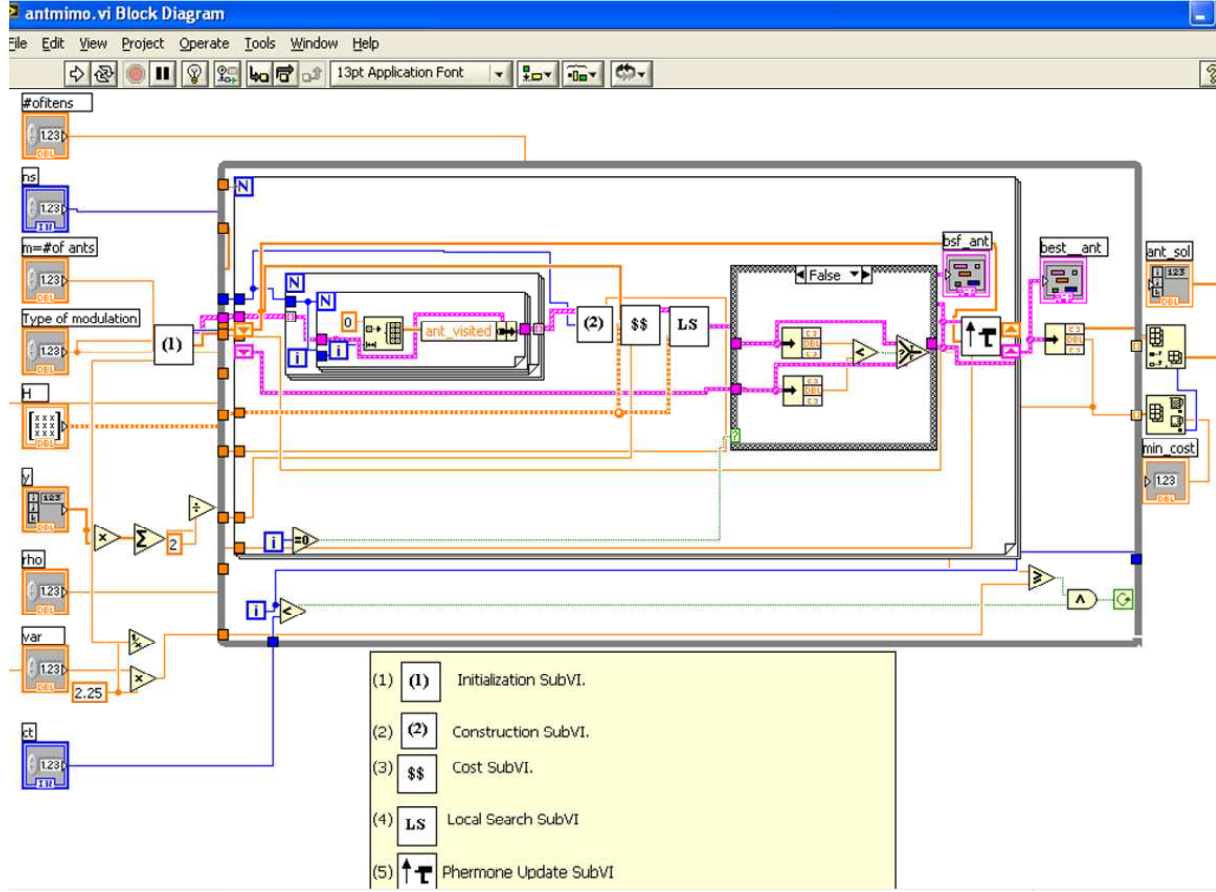


Fig. 33. Block diagram of the AntMIMO VI.

In this subVI, the pheromone trails for each child of $\text{ant}[k].\text{visited}[\text{index}]$ are retrieved and used to calculate the selection probabilities; the variable index indicates at which level of the tree the ants are currently residing. An ant chooses which next level node to include in its solution based on the roulette wheel selection strategy; the ant's solution array is updated with the new solution element x_i^k and the ants' visited nodes array is also updated with the selected node's location information .

- Cost Evaluation VI

This VI calculates the cost function $f(x^k) = ||y - Hx^k||^2$ for all constructed ants'

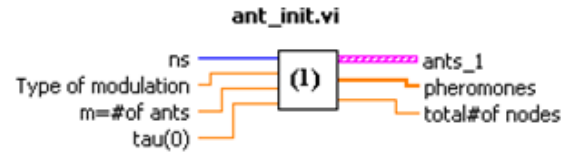


Fig. 34. Schematic representation of the initialization VI.

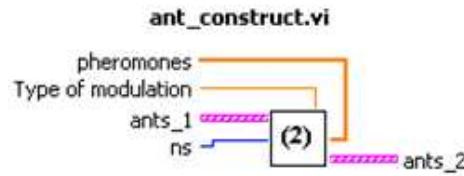


Fig. 35. Schematic representation of the construction VI.

solutions in a single iteration.

- Local Search VI

This VI implements the 1-opt local search procedure described in Algorithm 5.1 and outputs the ant structure with the minimum cost function for this iteration, best-it-ant.

- Update Pheromones VI

This VI updates the pheromone array according to the update rules of (5.4,5.5). The updated pheromone array reflects the search experience of the ants during run-

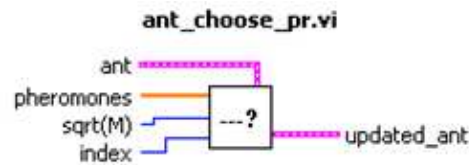


Fig. 36. Schematic representation of the ant-choose-pr subVI.



Fig. 37. Schematic representation of the cost evaluation VI.

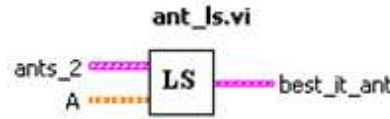


Fig. 38. Schematic representation of the local search VI.

time and is used to concentrate the search in promising regions of the search space. Moreover, we restrict the pheromone values to be greater than a small positive constant τ_{min} , so that each node has a non-zero probability of being selected.

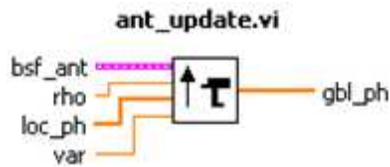


Fig. 39. Schematic representation of the pheromone update VI.

VITA

Dana N. Jaber received her Bachelor of Engineering degree in Communications and Electronics from Beirut Arab University, Lebanon in 2004. She joined the Wireless Communications and Signal Processing group with the Electrical and Computer Engineering Department at Texas A&M University in August 2004, and she received her Master of Science degree in August 2006. Her research interests include detection algorithms for multi-input multi-output wireless communication systems and biometric signal processing.

Ms. Jaber can be reached at Ericsson Inc., 6300 Legacy Drive, Plano, TX 75024. Her email address is djaber@ece.tamu.edu.